



Titre: Transformées reliées à celle de Fourier pour l'analyse spectrale et la
Title: compression des données

Auteur: Zaher Dannawi
Author:

Date: 2007

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Dannawi, Z. (2007). Transformées reliées à celle de Fourier pour l'analyse
Citation: spectrale et la compression des données [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7968/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7968/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

TRANSFORMÉES RELIÉES À CELLE DE FOURIER
POUR L'ANALYSE SPECTRALE
ET LA COMPRESSION DES DONNÉES

ZAHER DANNAWI
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENSION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
AVRIL 2007



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-29224-2

Our file Notre référence

ISBN: 978-0-494-29224-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

TRANSFORMÉES RELIÉES À CELLE DE FOURIER
POUR L'ANALYSE SPECTRALE
ET LA COMPRESSION DES DONNÉES

Présenté par : DANNAWI Zaher

En vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. CARDINAL Christian, Ph.D., président

M. CORINTHIOS J. Michael, Ph.D., membre et directeur de recherche

M. BRAULT Jean-Jules, Ph.D., membre

À mes parents ...

À mes frères...

Remerciement

Tout d'abord, je voudrais remercier mon directeur de recherche, le professeur Michael J. Corinthios pour la qualité de son encadrement , son encouragement et son support moral tout au long de ce mémoire.

Je remercie le Ministère de l'Éducation et de l'Enseignement Supérieur du Liban et plus particulièrement Madame Bouchra Bagdady Adra pour leurs soutien et aide pendant mes études supérieurs.

Enfin des remerciements plus personnels à mes parents et à mes frères qui m'ont soutenu et encouragé dans ce voyage éducatif, et je termine par un grand Merci à mon amie Dory Azar qui sans lui mon séjour à Québec aurait été plus difficile.

Résumé

Des transformées reliées à la Transformée Discrète de Fourier sont analysées dans cette recherche. L'analyse comprend une étude des champs de définition de ces transformées au domaine des signaux unidimensionnels ainsi que bidimensionnels. Une discussion de leurs propriétés et une comparaison des spectres qu'elles produisent par leur application à plusieurs classes de signaux sont présentées. Les transformées examinées sont la Transformée Discrète en Cosinus, la Transformée Discrète de Hartley, la Transformée Discrète de Walsh Hadamard, la Transformée Discrète de Fourier ainsi que celle relativement récente, la Transformée Discrète Fractionnelle de Fourier.

Le rôle de la transformée de Fourier dans la reconnaissance de la parole est étudié et discuté. Les limitations de la transformée traditionnelle de Fourier dans ce domaine sont démontrées. Des solutions qui résolvent ces contraintes, à l'aide des transformées reliées à Fourier sont présentées.

La propriété de compacité d'énergie dans la compression d'images de ces transformées est comparée en examinant les résultats obtenus par leurs applications à un ensemble d'images qui contient une variété de scènes. La qualité de l'image reconstruite est quantifiée par des paramètres communs dans le domaine de compression d'images.

L'étude montre que la transformée discrète en cosinus fournit le plus haut niveau de compacité d'énergie suivie par la transformée discrète de Hartley. La performance de la transformée de Fourier se retrouve entre cette paire de transformées et celle qui contient la Transformées Fractionnelles de Fourier et la Transformée de Walsh Hadamard.

Notre analyse de la Transformée de Walsh Hadamard a montré que sa performance ne se dégrade pas autant que les autres, lorsqu'une image avec un grand niveau de variance entre les pixels adjacents est traitée.

Abstract

A family of Fourier-related transforms are analysed in this research. In this analysis, the one-dimensional and the two-dimensional definitions of each transform are presented, their properties are discussed and the spectral results of their application on a set of signals are compared. The particular transforms are the Discrete Cosine Transform, the Discrete Hartley Transform, the Discrete Walsh Hadamard Transform, the Discrete Fourier Transform and the relatively new Discrete Fractional Fourier Transform.

The application of the Fourier Transform in the speech recognition is analysed. The limitations of the traditional Fourier transform are shown. Solutions of those limitations by some Fourier-related transforms are presented.

The property of the energy compactness of these transforms which is used in image compression is compared by examining the results obtained from a set of images of several types of scenes. Well known parameters in the image compression were used to measure the quality of the reconstructed images.

It is found that the Discrete Cosine Transform provides the highest energy compactness followed by the Discrete Hartley Transform. The performance of the Discrete Fourier Transform is shown to be less effective than the Cosine and Hartley transforms but better than the Fractional and Walsh Hadamard.

The performance of Walsh Hadamard transform is shown to degrade to a lesser degree than the others when an image with high inter-pixel change is analysed.

Table des Matières :

Remerciement.....	v
Résumé.....	vi
Abstract.....	vii
Table des Matières :.....	viii
Liste des Tableaux :.....	xi
Liste des Figures :.....	xii
Liste des Sigles et Abréviations :	xiv
1 Introduction	1
1.1 Organisation du mémoire:.....	2
2 Transformée Discrète de Fourier (TDF)	4
2.1 Définition :.....	4
2.2 Algorithmes Rapides pour la TDF :	4
2.3 Transformée Discrète bidimensionnelle de Fourier :.....	10
2.4 Propriétés de la Transformée Discrète de Fourier:	10
2.5 Orthogonalité de la Transformée Discrète de Fourier :	14
2.6 Interprétation en algèbre linéaire :	15
2.7 Application de la TDF à une classe de signaux :	15
3 Transformée Discrète en Cosinus (TDC)	20
3.1 Définition :.....	20
3.2 Algorithmes Rapides pour la TDC :	22
3.3 Transformée Discrète Bidimensionnelle en Cosinus :.....	23
3.4 Propriétés de la Transformée Discrète en Cosinus:.....	24
3.5 Orthogonalité de la Transformée Discrète en Cosinus :	24
3.6 Interprétation en algèbre linéaire :	25
3.7 Application de la TDC à une classe de signaux :	26
4 Transformée Discrète de Hartley (TDH).....	30
4.1 Définition :.....	30
4.2 Algorithme Rapide pour la TDH :	30
4.3 Transformée Discrète Bidimensionnelle de Hartley :	31

4.4	Propriétés de la Transformée Discrète de Hartley :	31
4.5	Orthogonalité de la Transformée Discrète de Hartley :	33
4.6	Interprétation en algèbre linéaire :	34
4.7	Application de la TDH à une classe des signaux :	35
5	Transformées Discrète de Walsh Hadamard (TDWH)	39
5.1	Définition :	39
5.1.1	Transformée de Walsh Hadamard ordre naturel :	39
5.1.1.1	Définition exponentielle de l'ordre naturel :	40
5.1.2	Transformée de Walsh Hadamard ordre dyadique :	41
5.1.2.1	Définition exponentielle de l'ordre dyadique :	42
5.1.3	Transformée de Walsh Hadamard ordre de séquence :	42
5.1.3.1	Définition exponentielle de l'ordre de séquence :	44
5.2	Algorithmes Rapides pour la TDWH :	45
5.3	Transformée de Walsh Généralisée et Algorithmes Rapides :	48
5.3.1	Transformée de Walsh Généralisée :	48
5.3.2	Factorisation pour un algorithme optimal :	50
5.4	Transformée Discrète Bidimensionnelle de Walsh Hadamard :	52
5.5	Propriétés de la Transformée Discrète de Walsh Hadamard :	53
5.6	Orthogonalité de la Transformée de Walsh Hadamard :	54
5.7	Application de la TDWH (ordre naturel) à une classe des signaux : ...	54
6	Transformée Discrète Fractionnelle de Fourier (TDFrF)	59
6.1	Définition :	59
6.1.1	Transformée Discrète Fractionnelle de Fourier (TDFrF) :	61
6.2	Transformée Discrète Bidimensionnelle Fractionnelle de Fourier	64
6.3	Propriétés de la Transformée Discrète Fractionnelle de Fourier :	65
6.4	Orthogonalité de la Transformée Discrète Fractionnelle de Fourier ...	66
6.5	Application de la TDFrF à une classe de signaux :	67
7	Application à la Reconnaissance de la parole	73
7.1	Système de la reconnaissance de la parole :	73
7.2	Transformée de Fourier dans la reconnaissance de la parole :	75

8	Application à la compression des images	80
8.1	Introduction	80
8.2	Description de l'Algorithme :	83
8.3	Méthode Expérimentale :	85
8.4	Résultat et Analyse :	87
8.5	Discussion:.....	90
9	Conclusion	92
10	Références	93
11	Annexe#1: Programme du Matlab	101

Liste des Tableaux :

Tableau 2. 1: Propriétés de la Transformée Discrète de Fourier.....	13
Tableau 2. 2 : Classe de signaux	16
Tableau 4. 1: Propriétés de la Transformée Discrète de Hartley.....	33
Tableau 8. 1: Valeurs de l'erreur lorsque 20% des composantes sont employées.....	88

Liste des Figures :

Figure 2. 1: machine asymétrique avec base=4 (radix-4)	8
Figure 2. 2: processeur rapide (High-Speed) à base =4	9
Figure 2. 3: Transformation d'un signal d'échelon par la TDF.....	16
Figure 2. 4: Transformation d'un signal rectangulaire par la TDF.	16
Figure 2. 5: Transformation d'un signal sinusoidal par la TDF.	17
Figure 2. 6: Transformation d'un signal exponentiel par la TDF.....	17
Figure 2. 7: Transformation d'un signal triangulaire par la TDF.	18
Figure 2. 8: Transformation d'un signal de rampe par la TDF.	18
Figure 2. 9: Résultat de la transformation d'une image par la TDF.	19
Figure 3. 1: Transformation d'un signal d'échelon par la TDC.....	26
Figure 3. 2: Transformation d'un signal rectangulaire par la TDC.....	27
Figure 3. 3: Transformation d'un signal sinusoidal par la TDC.....	27
Figure 3. 4: Transformation d'un signal exponentiel par la TDC.	28
Figure 3. 5: Transformation d'un signal triangulaire par la TDC.	28
Figure 3. 6: Transformation d'un signal de rampe par la TDC.....	29
Figure 3. 7: Résultat de la transformation d'une image par la TDC.....	29
Figure 4. 1: Transformation d'un signal d'échelon par la TDH.....	35
Figure 4. 2: Transformation d'un signal rectangulaire par la TDH.....	35
Figure 4. 3: Transformation d'un signal sinusoidal par la TDH.....	36
Figure 4. 4: Transformation d'un signal exponentiel par la TDH.	36
Figure 4. 5: Transformation d'un signal triangulaire par la TDH.	37
Figure 4. 6: Transformation d'un signal de rampe par la TDH.....	37
Figure 4. 7: Résultat de la transformation d'une image par la TDH.....	38
Figure 5. 1: Conversion d'ordre normal à ordre bit-reversé.....	41
Figure 5. 2: Conversion d'ordre binaire à ordre de gray.....	43
Figure 5. 3: Transformation d'un signal d'échelon par la TDWH.	55
Figure 5. 4: Transformation d'un signal rectangulaire par la TDWH.....	55
Figure 5. 5: Transformation d'un signal sinusoidal par la TDWH.	56
Figure 5. 6: Transformation d'un signal exponentiel par la TDWH.	56

Figure 5. 7: Transformation d'un signal triangulaire par la TDWH.....	57
Figure 5. 8: Transformation d'un signal de rampe par la TDWH.	57
Figure 5. 9: Résultat de la transformation d'une image par la TDWH.	58
Figure 6. 1: Domaine de la transformée fractionnelle d'ordre α	60
Figure 6. 2: Transformation d'un signal d'échelon par la TDFrF.	68
Figure 6. 3: Transformation d'un signal rectangulaire par la TDFrF.	68
Figure 6. 4: Transformation d'un signal sinusoidal par la TDFrF.....	69
Figure 6. 5: Transformation d'un signal exponentiel par la TDFrF.	70
Figure 6. 6: Transformation d'un signal triangulaire par la TDFrF.....	70
Figure 6. 7: Transformation d'un signal de rampe par la TDFrF.....	71
Figure 6. 8: Résultat de la transformation d'une image par la TDFrF.....	72
Figure 7. 1: Diagramme Fonctionnel général d'un système de reconnaissance de la parole.....	73
Figure 7. 2 : Signal vocal d'une personne.	74
Figure 7. 3 : Performance de la transformée de Fourier sur des signaux stationnaire et non stationnaire.....	76
Figure 7. 4: Fenêtre de Hamming et sa transformée de Fourier.	78
Figure 8. 1: Image de test#.....	85
Figure 8. 2: Image de test#2	86
Figure 8. 3: l'erreur (PSNR) sur pourcentage des composants de transformées pour l'image de test#1.....	88
Figure 8. 4: l'erreur (PSNR) sur pourcentage des composants de transformées pour l'image de test#2.....	89

Liste des Sigles et Abréviations :

ADN	Acide Désoxyribo-Nucléique
TDF	Transformée Discrète de Fourier
TDWH	Transformée Discrète de Walsh Hadamard
TDC	Transformée Discrète en Cosinus
TDH	Transformée Discrète de Hartley
TDFrF	Transformée Discrète fractionnelle de Fourier
$X(e^{j\Omega})$	Transformée de Fourier
$x[n]$	Vecteur de N point
Ω	Fréquence angulaire
f	Fréquence linéaire
$x_p[n]$	Résultat de la transformée inverse de Fourier
$X[k]$	Transformée Discrète de Fourier de $x[n]$
$x^*[n]$	Conjuguée de $x[n]$
$\text{Re}\{ \}$	Partie réelle
$\text{Im}\{ \}$	Partie imaginaire
\otimes	Convolution
F	Matrice équivalente à la transformée de Fourier
\wedge	Signal triangulaire
$X_c[k]$	Transformée Discrète en cosinus de $x[n]$
$X_c^I[k]$	Transformée Discrète en cosinus de $x[n]$ type I
$X_c^{II}[k]$	Transformée Discrète en cosinus de $x[n]$ type II
$X_c^{III}[k]$	Transformée Discrète en cosinus de $x[n]$ type III
$X_c^{IV}[k]$	Transformée Discrète en cosinus de $x[n]$ type IV
$X_s[k]$	Transformée Discrète en sinus de $x[n]$
D	Matrice équivalente à la transformée de cosinus

d_{nk}	Composantes de la matrice D
$H[k]$	Transformée de Hartley
$H_x^p[k]$	Partie paire de la transformée de Hartley
$H_x^i[k]$	Partie impaire de la transformée de Hartley
H	Matrice équivalente à la transformée de Hartley
h_{nk}	Composantes de la matrice H
$H_{nat}()$	Matrice de la transformée de Walsh Hadamard ordre naturel
$H_{dyad}()$	Matrice de la transformée de Walsh Hadamard ordre dyadique
$H_{seq}()$	Matrice de la transformée de Walsh Hadamard ordre séquence
$B_x[k]$	Transformée de Walsh Hadamard de $x[n]$ ordre naturel
$P_x[k]$	Transformée de Walsh Hadamard de $x[n]$ ordre dyadique
$W_x[k]$	Transformée de Walsh de $x[n]$ ordre séquence
\mathbb{F}	Opérateur de la transformée de Fourier.
$K_\alpha(t, u)$	Fonction de base de la transformée fractionnelle de Fourier
$X_{(p)}[m]$	Transformée Fractionnelle de $x[n]$ d'ordre p
$F_{(p)}$	Matrice de la transformée fractionnelle d'ordre p
$F_{(p)}^{-1}$	Matrice de la transformée fractionnelle inverse d'ordre p
$s[n]$	Signal de la parole
$w[n]$	Signal de la fenêtre
MD	Modulation de delta
MCID	Modulation du codage impulsion de delta
TKL	Transformée de Karhunen Loève
$O()$	Nombre de opération
eqm	Erreur quadratique moyenne

1 Introduction

La Transformée de Fourier et, en particulier, la Transformée Discrète de Fourier (TDF) est un outil majeur pour l'analyse spectrale des signaux. Elle a démontré qu'elle peut être indispensable dans presque tous les domaines scientifiques. Par exemple, en économie la transformée de Fourier a aidé les analystes à détecter les fluctuations périodiques du prix d'une action [1]. En Biologie, plusieurs chercheurs [2, 3] l'utilisent comme un moyen d'identifier le codage des gènes de protéine de l'ADN (Acide Désoxyribo-Nucléique).

La transformée de Fourier est de grande importance car plusieurs autres transformées en dépendent. Parmi ces transformées qui sont bien connues dans le domaine de traitement des signaux, on peut citer la Transformée Discrète de Walsh Hadamard (TDWH), la Transformée Discrète en Cosinus (TDC), la Transformée Discrète de Hartley (TDH) et la Transformée Discrète Fractionnelle de Fourier (TDFrF).

La Transformée de Chrestenson [4], un lien entre la Transformée Discrète de Walsh Hadamard (TDWH) et la Transformée de Fourier, est une forme généralisée de ces deux dernières. La Transformée Discrète de Cosinus (TDC) qui était introduite par Ahmed et al en 1974 [5] n'est que la forme numérique de la transformée de Fourier en cosinus [6].

La Transformée Discrète de Hartley (TDH) est une transformée directement reliée à la Transformée Discrète de Fourier (TDF). Sa distinction principale de la TDF se situe au niveau des entrées/sorties. Elle transforme les entrées réelles en des sorties réelles, sans l'intervention d'aucun calcul complexe. La forme continue de cette transformée était introduite par R.V. Hartley en 1942 [7]. Ce n'est qu'à 1983 que la forme discrète est introduite par R. N. Bracewell [8, 9, 10].

La Transformée Discrète Fractionnelle de Fourier (TDFrF) fut introduite dans la littérature mathématique dans les années 80 [11, 12]. Les chercheurs ont commencé à l'utiliser comme une technique de traitement des signaux dans les

années 90, par Almeida et al [13, 14]. Le domaine de la transformation est une fraction de celle du domaine de Fourier.

Les objectifs de cette recherche sont de faire une étude comparative des transformées reliées à celle de Fourier, de présenter leurs propriétés et de formaliser des transformées moins connues telle que Transformée Discrète en cosinus, Transformée Discrète de Hartley, Transformée de Walsh Généralisée et Transformée Discrète Fractionnelle de Fourier.

1.1 Organisation du mémoire:

Au chapitre 2, la Transformée Discrète de Fourier (TDF) est présentée. Une discussion de ses propriétés, ses algorithmes rapides et une démonstration de son orthogonalité sont faites. Ceci est suivi par une analyse expérimentale qui consiste en une application de cette transformée à une classe des signaux unidimensionnelle et bidimensionnelle.

Au chapitre 3, la Transformée Discrète en Cosinus (TDC) est traitée. La famille de cette transformée et leurs inverses sont explicitement décrits avec une discussion sur leurs algorithmes rapides. Suite à cette étude, quelques propriétés, y compris l'orthogonalité, seront discutées. Une analyse expérimentale de la transformée est faite par le calcul et l'affichage des spectres obtenus lors de son application à une classe des signaux.

Dans le chapitre 4, la Transformée Discrète de Hartley (TDH) est présentée. Ses propriétés et en particulier l'orthogonalité sont discutées. La même analyse expérimentale effectuée pour la Transformée Discrète de Fourier et de celle en Cosinus est faite pour la transformée de Hartley.

Au chapitre 5, la Transformée Discrète de Walsh Hadamard (TDWH) est analysée. On présente les trois ordres de cette transformée, l'ordre naturel, l'ordre dyadique et l'ordre de séquence. Pour chaque ordre, on fournit deux définitions : une matricielle et une exponentielle. Cela est suivi par une formalisation de la Transformée de Walsh Généralisée, une présentation de quelques algorithmes rapides et une discussion de quelques propriétés comme la convolution dyadique et l'orthogonalité.

L'analyse expérimentale faite dans les chapitres précédents est encore présentée pour cette transformée.

Dans le chapitre 6, la Transformée Discrète Fractionnelle de Fourier est analysée. Dans cette analyse, la forme continue est décrite et ses plusieurs formes numériques sont discutées et comparées. Une discussion de quelques propriétés de base et une démonstration de l'orthogonalité sont faites. Suite à cette étude, une analyse expérimentale qui consiste en une programmation de cette transformée à l'aide du logiciel Matlab est effectuée. Les résultats de son application à une classe des signaux pour différentes valeurs d'ordre α sont affichés.

Au chapitre 7, l'application de ces transformées dans le domaine de la reconnaissance de la parole est discutée. La discussion commence par une brève présentation du système de la reconnaissance. Les limitations sur la Transformée Discrète de Fourier sont démontrées. Cela est suivi par des solutions qui résolvent ces contraintes à l'aide de quelques transformées reliées à Fourier.

Au chapitre 8, l'application de ces transformées dans la compression des images est étudiée. Cette étude commencera par une introduction sur les algorithmes de compression sans et avec pertes d'information. Ceci est suivi par une description des algorithmes de compression de ces transformées. La méthode expérimentale adoptée dans cette analyse et les résultats obtenus sont présentés et discutés. Suite à ce travail une discussion générale est faite.

Au chapitre 9, une conclusion présente une énumération de travaux présentés dans ce mémoire avec un bref résumé des résultats expérimentaux obtenus.

2. Transformée Discrète de Fourier (TDF)

2.1.Définition :

La transformée de Fourier $X(e^{j\Omega})$ d'une séquence $x[n]$ est donnée par

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{n=+\infty} x[n] e^{-jn\Omega} \quad (2.1)$$

$X(e^{j\Omega})$ est une fonction périodique de 2π en Ω . Alors, elle peut donc être développée en une série de Fourier. En effet, les coefficients de la série de Fourier sont les coefficients de $x[n]$. Ces coefficients sont donnés par

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{j\Omega}) e^{jn\Omega} d\Omega \quad (2.2)$$

La Transformée Discrète de Fourier (TDF) est déduite de l'équation (2.1) lorsque le calcul est fait pour des valeurs de Ω données par

$$\Omega_k = k\Delta\Omega \quad (2.3)$$

avec $k = 0, 1, 2, 3, \dots, N-1$, des valeurs équidistantes de Ω tel que $\Delta\Omega = 2\pi/N$ et si $x[n]$ est de durée finie $n = 0, 1, 2, 3, \dots, N-1$

$$TDF\{x[n]\} = X(e^{j\Omega_k}) = X(e^{j2\pi k/N}) = X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \quad (2.4)$$

La Transformée Discrète de Fourier Inverse (TDFI) est donnée par

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N} \quad (2.5)$$

2.2.Algorithmes Rapides pour la TDF :

Le calcul de cette transformée pour une séquence de N points exige $O(N^2)$ opérations. Ces opérations sont faites pour des valeurs complexes. Des algorithmes ont été introduits pour réduire le nombre d'opérations en donnant les mêmes résultats. La plupart de ces algorithmes basent sur la factorisation de la matrice de la Transformée Discrète de Fourier (TDF). L'algorithme le plus commun est celle de Cooley-Tukey qui base sur l'algorithme 'Divide and Conquer'. L'idée de cet

algorithme est de diviser la transformée en deux parties de taille $N/2$ à chaque itération. Cette procédure est connue sous le nom 'Transformée rapide avec Radix-2' et la complexité du calcul sera égale à $O(N \log_2 N)$. Dans une autre classe, des algorithmes pour la factorisation de la matrice ont été introduites pour désigner des processeurs spécialisés de la Transformée Discrète de Fourier Rapide (TDFR). Parmi ces algorithmes celle de [15] qui a proposé une factorisation de la matrice de la Transformée Discrète de Fourier à l'aide de la multiplication Kronecker $(\cdot \times \cdot)$ comme c'est présenté par l'équation suivante :

$$T_{N/k} = P_{N/k} (T_{N/2k} \times I_2) D_{N/k} (I_{N/2k} \times T_2) \quad (2.6)$$

où $k = 2^i$; $i = 0, 1, 2, 3, \dots, n-1$, $T_{N/k}$ présente la matrice de la transformée pour N/k points, $T_{N/2k}$ la matrice de la transformée pour $N/2k$ points, $P_{N/k}$ la matrice de mélange parfaite, $D_{N/k}$ est une matrice quasi-diagonale $(I_{N/2k}, K_k)$, avec $I_{N/2k}$ la matrice d'identité de dimension $N/2k$, K_k donné par

$$K_k = \text{diag}(0, k, 2k, 3k, \dots) \quad (2.7)$$

et T_2 donnée par l'équation suivante

$$T_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.8)$$

Dans l'équation (2.6) $T_{N/2k}$ peut être décomposé d'une même façon. Cette procédure est refaite jusqu'à la matrice de la transformée dépend seulement de T_2 :

$$T_N = P_N [\{ P_{N/2} [\{ \dots P_{N/k} [\{ \dots [\{ P_4 (T_2 \times I_2) D_4 (I_2 \times T_2) \} \times I_2 \dots] \times I_2 \} D_{N/K} (I_{N/2k} \times T_2) \dots] \times I_2 \} D_{N/2} (I_{N/4} \times T_2) \} \times I_2 \} D_N (I_{N/2} \times T_2) \quad (2.9)$$

La factorisation donnée par l'équation (2.9) est ensuite appliquée dans deux algorithmes pour produire des processeurs séquentiels et rapides. Le premier effectue la permutation d'une façon antérieure (Post-Permutation) et le deuxième serre la permutation dans les itérations (Ordered-Input Ordered-Output) et par cela cet algorithme sera plus convenable aux machines.

Algorithme avec permutation antérieure (Post-Permutation) :

Une relation entre la multiplication matricielle et le produit Kronecker est donnée par l'équation suivante :

$$(ABC...) \times I = (A \times I)(B \times I)(C \times I)... \quad (2.10)$$

En appliquant l'équation (2.10) à (2.9) on obtient

$$\begin{aligned} T_N = & P_N (P_{N/2} \times I_2) \dots (P_{N/k} \times I_k) \dots (P_4 \times I_{N/4}) \\ & \cdot (T_2 \times I_{N/2}) (D_4 \times I_{N/4}) (I_2 \times T_2 \times I_{N/4}) \dots \\ & \cdot (D_{N/k} \times I_k) (I_{N/2k} \times T_2 \times I_k) \dots \\ & \cdot (D_{N/2} \times I_2) (I_{N/4} \times T_2 \times I_2) D_N (I_{N/2} \times T_2) \end{aligned} \quad (2.11)$$

La permutation de mélange parfaite possède la propriété suivant

$$P^{-1} (I_{N/2} \times T_2) P = I_{N/4} \times T_2 \times I_2 \quad (2.12)$$

Si on écrit

$$S = (I_{N/2} \times T_2) \quad (2.13)$$

alors d'une façon générale (2.12) sera égale à

$$P^{-i} S P^i = I_{N/2^{i+1}} \times T_2 \times I_{2^i} \quad (2.14)$$

et si on écrit

$$Q_i = P_{N/2^{i-1}} \times I_{2^{i-1}} \quad (2.15)$$

$$M_i = D_{N/2^{i-1}} \times I_{2^{i-1}} \quad (2.16)$$

l'équation (2.11) peut être présentée maintenant après l'introduction des équations (2.13), (2.14) et (2.15) :

$$T_N = Q_1 Q_2 \dots Q_{n-1} P S P M'_{n-1} \dots S P M'_2 S P M'_1 S \quad (2.17)$$

où

$$M'_i = P^{i-1} M_i P^{-(i-1)} \quad (2.18)$$

Finalement l'équation de l'algorithme est donnée par :

$$T_N = \prod_{m=1}^{n-1} Q_i \prod_{m=n}^1 (P M'_m S) \quad (2.19)$$

L'algorithme Entrée-Ordonnée Sortie-Ordonnée (Ordered-Input Ordered- Output) :

Pour inclure la permutation dans les itérations on utilise les propriétés suivantes :

$$P_k (A_{k/2} \times I_2) P_k^{-1} = I_2 \times A_{k/2} \quad (2.20)$$

et

$$P_k(ABC...)P_k^{-1} = (P_kAP_k^{-1})(P_kBP_k^{-1})(P_kCP_k^{-1})... \quad (2.21)$$

En appliquant ses propriétés à l'équation (2.9) on obtient :

$$\begin{aligned} T_N &= (I_{N/2} \times T_2)(I_{N/4} \times P_4)(I_{N/4} \times D_4)(I_{N/2} \times T_2)... \\ &\quad \cdot (I_k \times P_{N/k})(I_k \times D_{N/k})(I_{N/2} \times T_2)... \\ &\quad \cdot (I_2 \times P_{N/2})(I_2 \times D_{N/2})(I_{N/2} \times T_2)P_N D_N (I_{N/2} \times T_2) \end{aligned} \quad (2.22)$$

En utilisant l'équation (2.13) avec

$$P_i' = I_{2^{i-1}} \times P_{N/2^{i-1}} \quad (2.23)$$

et

$$\mu_i = I_{2^{i-1}} \times D_{N/2^{i-1}} \quad (2.24)$$

l'équation (2.22) sera écrite comme ci-dessous :

$$T_N = \prod_{m=n}^1 (P_m' \mu_m S) \quad (2.25)$$

Cet algorithme produit une machine câblée (Wired-in), sans adressage, qui effectue l'ordonnance de la sortie pendant les itérations du feedback et par cela les itérations pour la permutation antérieure sont éliminées.

Dans [16] une classe des machines a été introduite avec un grand ordre de parallélisme. Cette classe est basée sur des algorithmes qui sont la généralisation de celles présentées ci-dessus, en utilisant une base (radix) générale.

Selon cette généralisation, les équations (2.19) et (2.25) de l'algorithme avec permutation antérieure et l'algorithme avec permutation incluse dans les itérations respectivement sont données par :

$$T_N = \prod_{m=1}^{n-1} Q_i^{(r)} \prod_{m=1}^n (P_m^{(r)} M_m^{(r)} S^{(r)}) \quad (2.26)$$

et

$$T_N = \prod_{m=1}^n (P_m^{(r)} \mu_m^{(r)} S^{(r)}) \quad (2.27)$$

où r est la valeur de la base (radix).

La figure (2.1) [16], montre un diagramme fonctionnel d'une machine qui est la réalisation de ces deux algorithmes (base=4). La séquence de donnée (N-points)

qui peut être réécrite :

$$T_N = S_1^{(r)} \mu_1^{(r)} S_2^{(r)} \mu_3^{(r)} \dots S_{n-2}^{(r)} \mu_{n-1}^{(r)} S_{n-1}^{(r)} \mu_n^{(r)} S_n^{(r)} \quad (2.29)$$

où

$$\begin{aligned} S_{m-1}^{(r)} &= S^{(r)} P_m^{(r)}; \quad m = 2, 3, \dots, n \\ S_n^{(r)} &= S^{(r)} \end{aligned} \quad (2.30)$$

et

$$\mu_1 = I_N \quad (2.31)$$

Le nouveau opérateur $S_m^{(r)}$ combine les données séparées par au moins N/r^2 mots comme les équations suivantes le montrent :

$$\begin{aligned} S_{m-1}^{(r)} &= S^{(r)} P_m^{(r)} = (I_{N/r} \times T_r) P_m \\ &= P_m P_m^{-1} (I_{N/r} \times T_r) P_m \\ &= P_m (I_{N/r^2} \times T_r \times I_r) \end{aligned} \quad (2.32)$$

Alors l'équation (2.27) après l'application de modifications est donnée par

$$T_N = \prod_{m=1}^n (\mu_m^{(r)} S_m^{(r)}) \quad (2.33)$$

Basant sur cet algorithme rapide de l'équation (2.33), dans [58] une machine a été construite pour calculer la Transformée Discrète de Fourier avec une base égale à 4 (radix-4). Le calcul de la transformée de Fourier par cette machine était démontré qu'il prend moins de temps que par d'autres machines.

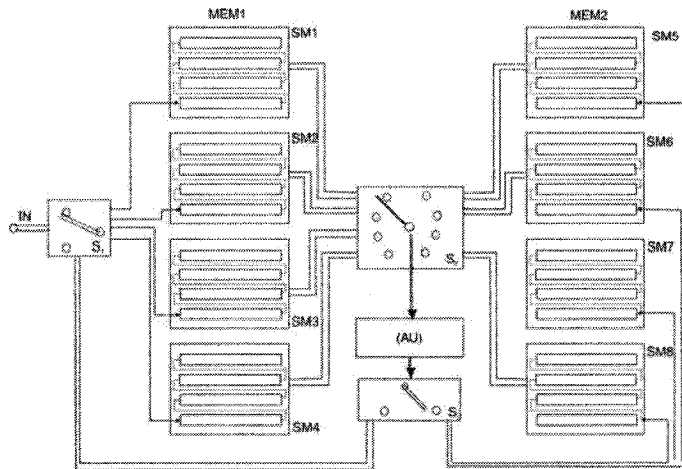


Figure 2. 2: processeur rapide (High-Speed) à base =4

La figure ci-dessus [17] montre la structure d'une machine avec base=4 (radix-4) qui est la réalisation de l'algorithme modifié. Cette machine possède deux mémoire MEM1 et MEM2, 'Arithmetic Unit (AU)' et trois commutateurs (S1, S2 et S3). Le AU calcule les deux opérateurs $S^{(r)}$ et μ_m . Cette machine calcule la Transformée Discrète de Fourier avec une haute vitesse (High-speed), $r+1=5$ plus rapide que celle de la figure (2.1). Ce gain de vitesse est obtenu par l'élimination de l'opération de permutation qui était faite pendant le feedback dans la figure (2.1).

2.3. Transformée Discrète bidimensionnelle de Fourier :

Pour le traitement des signaux bidimensionnelle comme les images $x[n_1, n_2]$, où x à $N \times N = N^2$ échantillons, la Transformée Discrète de Fourier prendra la forme suivante

$$TDF \{x[n_1, n_2]\} = X[k_1, k_2] = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x[n_1, n_2] e^{-j2\pi(n_1 k_1 + n_2 k_2)/N^2} \quad (2.35)$$

et la transformée inverse sera :

$$TDFI \{X[k_1, k_2]\} = x[n_1, n_2] = \frac{1}{N^2} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} X[k_1, k_2] e^{j2\pi(k_1 n_1 + k_2 n_2)/N^2} \quad (2.36)$$

2.4. Propriétés de la Transformée Discrète de Fourier:

Linéarité :

Si $x_1[n]$ et $x_2[n]$ sont deux signaux à durée limitée à N_1 et N_2 , alors la TFD de

$$x[n] = ax_1[n] + bx_2[n] \quad (2.37)$$

Est donnée par

$$X[K] = aX_1[K] + bX_2[K] \quad (2.38)$$

Où $X_1[k]$ et $X_2[k]$ sont les Transformées Discrète de Fourier de $x_1[n]$ et $x_2[n]$ respectivement.

Renversement temporel :

Si $x[n]$ est une séquence de N valeurs et sa Transformée Discrète de Fourier est $X[k]$.

La Transformée Discrète de Fourier de $x[-n]$ est donnée par

$$TDF\{x[-n]\} = X[-k] \quad (2.39)$$

Conjugaison :

Si $x^*[n]$ est la conjuguée d'une séquence complexe $x[n]$ qui a comme Transformée Discrète de Fourier $X[k]$, alors sa transformée est donnée par

$$TDF\{x^*[n]\} = X^*[-k] \quad (2.40)$$

Décalage temporel et fréquentiel :

Si on a une séquence périodique $x[n]$, de période N , qui est décalée de n_0 échantillons vers la droite, ce décalage est indiscernable d'un décalage $n_0 \bmod N$.

Alors la TDF du signal $y[n] = x[n - n_0]$ est donnée par

$$Y[k] = X[k] e^{j2\pi n_0 / N} \quad (2.41)$$

D'une même façon, si on a une séquence dans le domaine fréquentiel $X[k]$ qui est décalée de k_0 valeurs vers la droite, ce décalage est indiscernable d'un décalage $k_0 \bmod N$. La TDFI du signal $X[k - k_0]$ est donnée par

$$TDFI\{X[k - k_0]\} = x[n] e^{j2\pi k_0 / N} \quad (2.42)$$

Symétrie :

Si $x[n]$ est une séquence réelle dans l'intervalle $(0, N-1)$, alors on peut déduire que

$$\operatorname{Re}\{X[k]\} = \operatorname{Re}\{X[-k]\} \quad (2.43)$$

$$\operatorname{Im}\{X[k]\} = -\operatorname{Im}\{X[-k]\} \quad (2.44)$$

$$|X[k]| = |X[-k]| \quad (2.45)$$

Si $x[n] = x[-n]$, alors la partie imaginaire de la Transformée Discrète de Fourier de ce signal doit être égale à 0. Dans ce cas on a l'égalité suivante

$$X[k] = X^*[-k] \quad (2.46)$$

Convolution et corrélation :

Si $x[n]$ et $h[n]$ sont deux signaux réels tel que

$$TDF\{x[n]\} = X[k] \quad (2.47)$$

$$TDF\{h[n]\} = H[k] \quad (2.48)$$

et leurs convolution circulaire est donnée par

$$y[n] = (x[u] \otimes h[n])_N = \frac{1}{N} \sum_{u=0}^{N-1} x[u] h[n-u] \quad (2.49)$$

Alors (2.42) sera dans le domaine de Fourier

$$Y[k] = X[k]H[k] \quad (2.50)$$

Preuve :

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u] h[n-u] \exp(-j2\pi nk / N) \\ &= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u] h[n-u] \exp(-j2\pi(n-u+u)k / N) \\ &= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u] h[n-u] \exp(-j2\pi(n-u)k / N) \exp(-j2\pi uk / N) \\ &= \sum_{u=0}^{N-1} x[u] \exp(-j2\pi uk / N) \sum_{n=0}^{N-1} h[n-u] \exp(-j2\pi(n-u)k / N) \\ &= X[k]H[k] \end{aligned} \quad (2.51)$$

Si $x[n]$ et $h[n]$ sont deux signaux réels tel que

$$TDF\{x[n]\} = X[k] \quad (2.52)$$

$$TDF\{h[n]\} = H[k] \quad (2.53)$$

et leurs corrélation circulaire est donnée par

$$r_{xh}[n] = \frac{1}{N} \sum_{u=0}^{N-1} x[u] h[n+u] \quad (2.54)$$

Alors (2.47) sera dans le domaine de Fourier

$$R_{xh}[k] = X^*[k]H[k] \quad (2.55)$$

Preuve :

$$\begin{aligned}
R_{xh}[k] &= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u]h[n+u] \exp(-j2\pi nk / N) \\
&= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u]h[n+u] \exp(-j2\pi(n-u+u)k / N) \\
&= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} x[u]h[n-u] \exp(-j2\pi(n+u)k / N) \exp(-j2\pi(-u)k / N) \\
&= \sum_{u=0}^{N-1} x[u] \exp(-j2\pi(-u)k / N) \sum_{n=0}^{N-1} h[n+u] \exp(-j2\pi(n+u)k / N) \quad (2.56) \\
&= X^*[k]H[k]
\end{aligned}$$

Les propriétés discutées ci-dessus se résument dans le tableau (2.1).

Propriété	Fonction	Transformée de Fourier Discrète
	$x[n]$	$X[k]$
Linéarité	$x[n] = ax_1[n] + bx_2[n]$	$X[k] = aX_1[k] + bX_2[k]$
Renversement	$x[-n]$	$X[-k]$
Translation	$x[n - n_0]$	$X[k] e^{j2\pi n_0 / N}$
Modulation	$x[n] e^{j2\pi k_0 / N}$	$X[k - k_0]$
Convolution	$(x_1[n_1] \otimes x_2[n_2])_N$	$X_1[k_1] X_2[k_2]$
Corrélation	$(x_1[n_1] \oplus x_2[n_2])_N$	$X_1[k_1] X_2^*[k_2]$
Multiplication	$x_1[n_1] x_2[n_2]$	$\frac{1}{N} X_1[k_1] \otimes X_2[k_2]$
Changement d'échelle	$x[n/a]$	$ a X[ak]$
Conjugaison Complexe	$x^*[n]$	$X^*[-k]$
Symétrie Hermitienne	$x[n] \in \mathbb{R}$	$X[-k] = X^*[k]$

Tableau 2. 1: Propriétés de la Transformée Discrète de Fourier.

2.5. Orthogonalité de la Transformée Discrète de Fourier :

Le produit scalaire de deux vecteurs de N-dimensions est donné par l'équation suivante

$$\langle g_k, g_m \rangle = \sum_{n=0}^{N-1} g_{nk} g_{nm}^* \quad (2.57)$$

Les vecteurs g_k et g_m sont dites orthogonaux, si leur produit est égal à zéro lorsque $k \neq m$. Alors pour qu'une transformée soit orthogonale, ces vecteurs de base doivent obéir à l'équation suivante [6]

$$\langle g_k, g_m \rangle = \begin{cases} c, & k = m \\ 0, & k \neq m \end{cases} \quad (2.58)$$

où $c = cte \neq 0$.

La Transformée Discrète de Fourier décrite par l'équation (2.4) peut être représentée encore par

$$TDF\{x[n]\} = X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad (2.59)$$

où

$$W_N^{nk} = (e^{-j2\pi/N})^{nk} \quad (2.60)$$

Alors

$$\begin{aligned} \langle W_N^{nk_1}, W_N^{nk_2} \rangle &= \sum_{n=0}^{N-1} e^{-j2\pi nk_1/N} e^{j2\pi nk_2/N} \\ &= \sum_{n=0}^{N-1} e^{-j2\pi n(k_1 - k_2)/N} = \begin{cases} N, & k_1 = k_2 \\ 0, & k_1 \neq k_2 \end{cases} \end{aligned} \quad (2.61)$$

Il y a deux cas à étudier

1) $k_1 = k_2$

$$\langle W_N^{nk_1}, W_N^{nk_2} \rangle = \sum_{n=0}^{N-1} e^{-j2\pi n(0)/N} = \sum_{n=0}^{N-1} 1 = N \quad (2.62)$$

2) $k_1 \neq k_2$

$$\langle W_N^{nk_1}, W_N^{nk_2} \rangle = \sum_{n=0}^{N-1} e^{-j2\pi n(k_1 - k_2)/N} = \sum_{n=0}^{N-1} e^{-j2\pi n(k_3)/N} = 0 \quad (2.63)$$

2.6. Interprétation en algèbre linéaire :

La Transformée Discrète de Fourier peut être considérée comme une transformation linéaire appliquée à un vecteur $x[n]$ afin de donner le vecteur $X[k]$

$$X[k] = Fx[n] \quad (2.64)$$

Où F est une matrice avec les coefficients $f_{nk} = e^{-j2\pi nk/N}$

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} e^{-j2\pi 0.0/N} & e^{-j2\pi 0.1/N} & \dots & e^{-j2\pi 0.N-1/N} \\ e^{-j2\pi 1.0/N} & e^{-j2\pi 1.1/N} & \dots & e^{-j2\pi 1.N-1/N} \\ \vdots & \vdots & \vdots & \vdots \\ e^{-j2\pi N-1.0/N} & e^{-j2\pi N-1.1/N} & \dots & e^{-j2\pi N-1.N-1/N} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (2.65)$$

Ou bien

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & \dots & W_N^{N-1} \\ \vdots & \vdots & \dots & \vdots \\ W_N^0 & W_N^{N-1} & \dots & W_N^{N-1.N-1} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (2.66)$$

2.7. Application de la TDF à une classe de signaux :

La Transformée Discrète de Fourier et la transformée inverse sont programmées pour calculer et afficher les spectres d'une classe de signaux. En général, la programmation peut être faite par plusieurs logiciels. Le logiciel utilisé dans cette étude est celui de Matlab. La classe de signaux étudiée est présentée dans le tableau (2.2).

Signal	Équation du signal
signal d'échelon	$x[n] = u[n]$
signal rectangulaire	$x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$
signal sinusoidal	$x[n] = \sin(2\pi fn)$
signal exponentiel	$x[n] = \exp(-\alpha n)$

signal triangulaire	$x[n] = \Lambda_N$
signal de rampe	$x[n] = n$
signal bidimensionnel	$x[n] = \text{image}$ (figure 8.1)

Tableau 2. 2 : Classe de signaux

Signal d'échelon : $x[n] = u[n]$

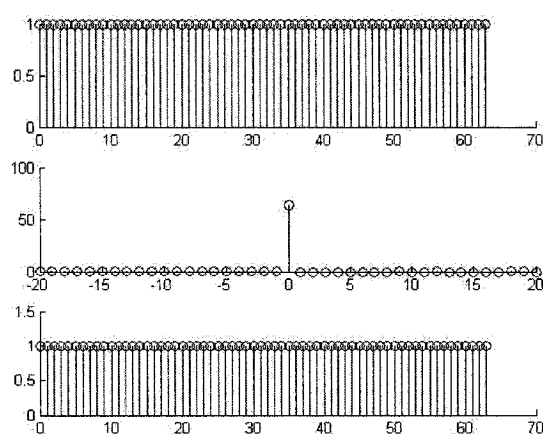


Figure 2. 3: Transformation d'un signal d'échelon par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal rectangulaire : $x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$

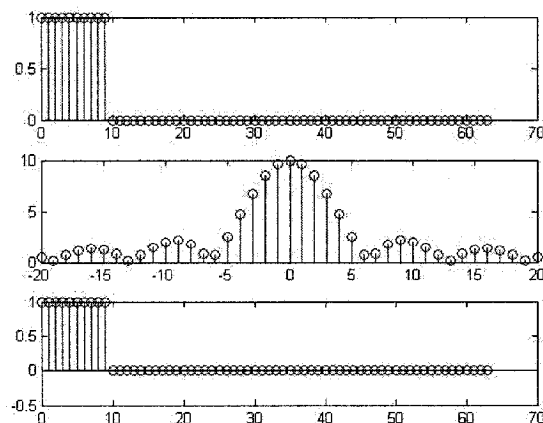


Figure 2. 4: Transformation d'un signal rectangulaire par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal sinusoïdal : $x[n] = \sin(2\pi fn)$

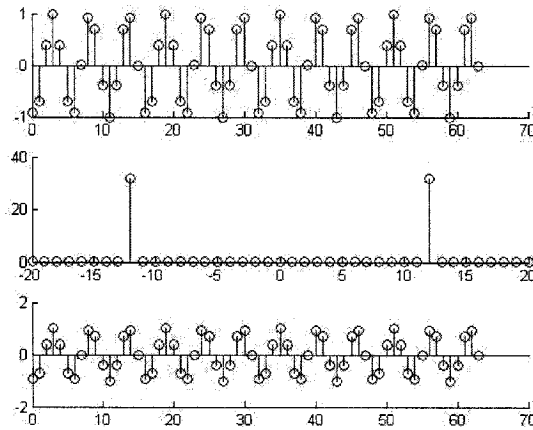


Figure 2.5: Transformation d'un signal sinusoïdal par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit

Signal exponentiel : $x[n] = \exp(-\alpha n)$

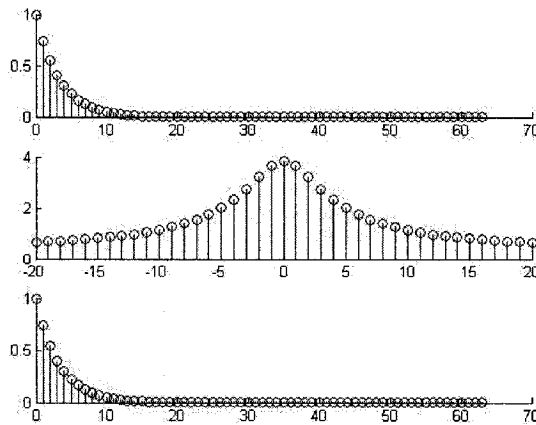


Figure 2.6: Transformation d'un signal exponentiel par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal triangulaire : $x[n] = \Lambda_N$

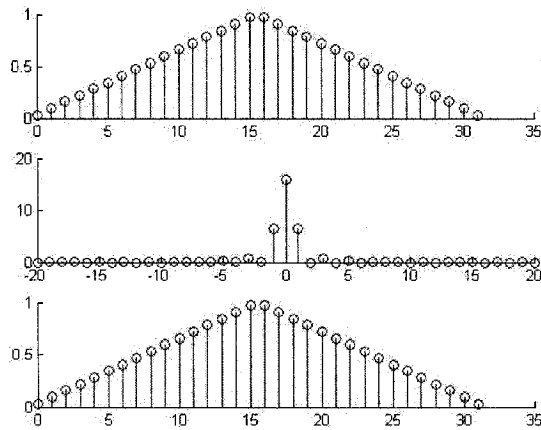


Figure 2. 7: Transformation d'un signal triangulaire par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal de rampe : $x[n] = n$

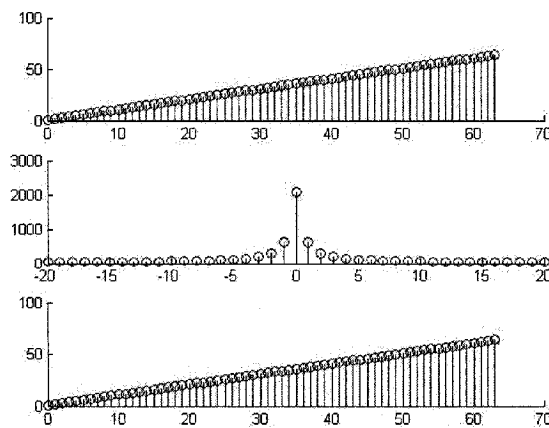


Figure 2. 8: Transformation d'un signal de rampe par la TDF.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal bidimensionnel : $x[n] = \text{image}$

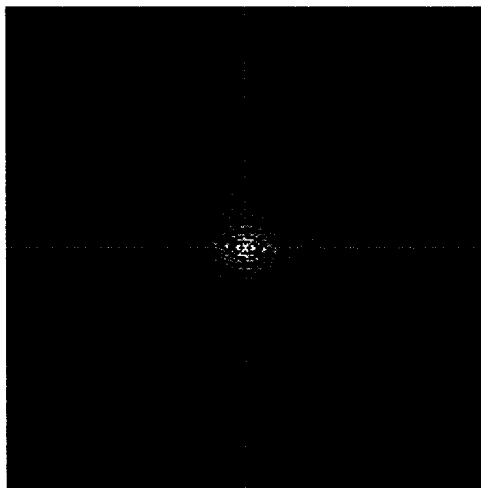


Figure 2. 9: Résultat de la transformation d'une image par la TDF.

3. Transformée Discrète en Cosinus (TDC)

3.1.Définition :

La Transformée Discrète en Cosinus (TDC), comme la Transformée de Fourier, transforme une fonction dans le domaine temporelle à une fonction dans le domaine fréquentielle. Elle a été tout d'abord introduite par Ahmed et al en 1974 [5] comme la forme numérique de la transformée de Fourier en cosinus [6].

Il y a plusieurs formes de cette transformée mais toutes transforment la fonction échantillonnée, représentée par un vecteur $x[n]$ de N valeurs réelles, en un vecteur $X[k]$ encore de N valeurs réelles dans le domaine fréquentielle.

Ces formes sont données par les équations suivantes :

TDC-I : Transformée Discrète en Cosinus de type I

Transformation directe :

$$X'_c[k] = \sqrt{\frac{2}{N}} w_k \sum_{n=0}^N w_n x[n] \cos\left[\frac{\pi nk}{N}\right] \quad k = 0, \dots, N \quad (3.1)$$

Transformation inverse :

$$x[n] = \sqrt{\frac{2}{N}} w_n \sum_{k=0}^N w_k X'_c[k] \cos\left[\frac{\pi nk}{N}\right] \quad n = 0, \dots, N \quad (3.2)$$

où

$$w_n = w_k = \begin{cases} \frac{1}{\sqrt{2}} \rightarrow \text{pour } t = 0, \text{ ou } N \\ 1 \rightarrow \text{autrement} \end{cases} \quad (3.3)$$

Cette forme est définie seulement pour les valeurs positives de $N > 2$.

TDC-II : Transformée Discrète en Cosinus de type II

Transformation directe :

$$X''_c[k] = \sqrt{\frac{2}{N}} w_k \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi}{2N} (2n+1)k\right] \quad k = 0, \dots, N-1 \quad (3.4)$$

Transformation inverse :

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} w_k X_c^{II}[k] \cos\left[\frac{\pi}{2N}(2n+1)k\right] \quad n = 0, \dots, N-1 \quad (3.5)$$

où w_k est définie dans (3.3).

Cette forme est en générale la plus utilisable par les chercheurs et par les logiciels comme Matlab. Elle est souvent désignée par « la transformée discrète en cosinus (TDC) ». La TDC-II est définie pour toutes les valeurs positives de N.

TDC-III : Transformée Discrète en Cosinus de type III

Transformation directe :

$$X_c^{III}[k] = \sqrt{\frac{2}{N}} \sum_{n=1}^{N-1} w_n x[n] \cos\left[\frac{\pi}{2N}n(2k+1)\right] \quad k = 0, \dots, N-1 \quad (3.6)$$

Transformation inverse :

$$x[n] = \sqrt{\frac{2}{N}} w_n \sum_{k=1}^{N-1} X_c^{III}[k] \cos\left[\frac{\pi}{2N}n(2k+1)\right] \quad n = 0, \dots, N-1 \quad (3.7)$$

où w_n est définie dans (3.3).

Cette forme est définie pour toutes les valeurs positives de N.

TDC-IV : Transformée Discrète en Cosinus de type IV

Transformation directe :

$$X_c^{IV}[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi}{4N}(2n+1)(2k+1)\right] \quad k = 0, \dots, N-1 \quad (3.8)$$

Transformation inverse :

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_c^{IV}[k] \cos\left[\frac{\pi}{4N}(2n+1)(2k+1)\right] \quad n = 0, \dots, N-1 \quad (3.9)$$

Cette forme est définie pour toutes les valeurs positives de N.

En examinant ces différentes formes, on réalise que la Transformée en Cosinus de type I et type IV ont des matrices involutives dont leurs inverses ne sont que les matrices de la transformation directe elle-même. La transformée inverse de type II est similaire à la transformation directe de type III et vice versa.

Dans ce qui suit dans notre étude sur la Transformée Discrète en Cosinus, on se concentrera seulement sur le type II et son inverse le type III. Elles sont désignées

tout simplement par « la Transformée Discrète en Cosinus » et « la Transformée Discrète en Cosinus Inverse » respectivement.

3.2.Algorithmes Rapides pour la TDC :

Initialement, les chercheurs ont concentrée dans leur étude à trouver un algorithme rapide pour cette transformée de calculer la TDC à travers la Transformée Discrète de Fourier Rapide (TDFR). Par exemple, Haralick [18] a présenté une approche de calcul de la TDC de N-points par deux N-points TDFR. Autres algorithmes ont été développés sans l'utilisation de la Transformée Discrète de Fourier. Ces algorithmes se basent sur la factorisation de la matrice da la Transformée Discrète en Cosinus. L'idée de la factorisation est de décomposer la matrice de la TDC en des matrices clairsemées ou répandues. En calculant ces nouvelles matrices, on obtient les mêmes résultats de la Transformée en Cosinus mais avec moins d'opérations arithmétiques. Wang [19] a proposé une factorisation de la matrice de la TDC comme c'est présenté ci-dessous :

$$C_N = P'_N \begin{bmatrix} C_{N/2} & 0 \\ 0 & R_{N/2}^* \end{bmatrix} B_N \quad (3.10)$$

où C_N la matrice de la Transformée Discrète en Cosinus type-II pour N-points, P'_N une matrice de permutation qui arrange les lignes pairs dans un ordre croissant dans la première partie et les lignes impaires dans un ordre décroissant dans la deuxième partie, $R_{N/2}^*$ obtenue en reversant les ordres des lignes et des colonnes de la matrice $R_{N/2}$ dont les éléments son donnés par l'équation suivante:

$$[R_N]_{i,k} = \cos \left[\frac{(2i+1)(2k+1)\pi}{4N} \right]; \quad k, i = 0, 1, 2, \dots, N-1 \quad (3.11)$$

et la matrice B_N peut être représenter par la matrice d'identité et la matrice d'identité opposée.

Exemple pour N=4 :

$$C_4 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} \quad (3.12)$$

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ C_4^1 & C_4^3 & C_4^3 & C_4^1 \\ C_8^3 & -C_8^1 & C_8^1 & -C_8^3 \\ C_8^1 & C_8^3 & -C_8^3 & -C_8^1 \end{bmatrix} \quad (3.13)$$

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ C_4^1 & C_4^3 & 0 & 0 \\ 0 & 0 & -C_8^1 & C_8^3 \\ 0 & 0 & C_8^3 & C_8^1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \quad (3.14)$$

$$C_4 = P_4' \begin{bmatrix} C_2 & 0 \\ 0 & R_2^* \end{bmatrix} B_4 \quad (3.15)$$

3.3. Transformée Discrète Bidimensionnelle en Cosinus :

La Transformée Discrète en Cosinus d'une image ou d'une matrice de $N \times N$ est donnée par l'équation suivante :

$$X_c[k_1, k_2] = \frac{2}{\sqrt{N_1 N_2}} w_{k_1} w_{k_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \cos \left[\frac{\pi}{2N_1} (2n_1 + 1)k_1 \right] \cos \left[\frac{\pi}{2N_2} (2n_2 + 1)k_2 \right] \quad (3.16)$$

et la transformée inverse bidimensionnelle par l'équation suivante :

$$x[n_1, n_2] = \frac{2}{\sqrt{N_1 N_2}} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} w_{k_1} w_{k_2} X_c[k_1, k_2] \cos \left[\frac{\pi}{2N_1} k_1 (2n_1 + 1) \right] \cos \left[\frac{\pi}{2N_2} k_2 (2n_2 + 1) \right] \quad (3.17)$$

Où w_k est définie par l'équation (3.3).

3.4. Propriétés de la Transformée Discrète en Cosinus:

Linéarité :

Si $x_1[n]$ et $x_2[n]$ sont deux signaux à durée limitée avec N_1 et N_2 , alors la TDC de

$$x[n] = ax_1[n] + bx_2[n] \quad (3.18)$$

est donnée par

$$X_c[k] = aX_{c1}[k] + bX_{c2}[k] \quad (3.19)$$

Changement d'échelle :

Si $x[n]$ est un signal numérique limité de N , où sa TDC est

$$TDC\{x[n]\} = X_c[k] \quad (3.20)$$

Alors la TDC du $x[an]$ est

$$TDC\{x[an]\} = X_c[k/a] \quad (3.21)$$

Décalage temporel :

Si on a une séquence périodique $x[n]$, de période N , qui est décalée d'un échantillon vers la droite, alors la TDC du signal $y[n] = x[n-1]$ est donnée par

$$\begin{aligned} Y_c[k] &= \cos(k\pi/N)X_c[k] + \sin(k\pi/N)X_s[k] \\ &+ \sqrt{\frac{2}{N}}w_k[(-1)^k x[N] - x[0]]\cos(k\pi/2N) \end{aligned} \quad (3.22)$$

où $X_c[k]$ est la transformée en cosinus de $x[n]$, $X_s[k]$ est la transformée en sinus de $x[n]$ et w_k est défini par l'équation (3.3).

Pour une liste complète de propriétés de cette transformée, y compris la convolution et la différence, veuillez vous référer aux articles [6, 21, 22].

3.5. Orthogonalité de la Transformée Discrète en Cosinus :

Pour que la transformée Discrète en Cosinus soit orthogonal, on doit démontrer que ces vecteurs de base obéissent à l'équation (2.58).

Les vecteurs de base de cette transformée ont la forme suivante :

$$\cos_N^{nk} = \sqrt{\frac{2}{N}} w_k \cos(\pi(n+1/2)k/N) \quad (3.23)$$

alors

$$\langle \cos_N^{nk_1}, \cos_N^{nk_2} \rangle = \frac{2w_{k_1}w_{k_2}}{N} \sum_{n=0}^{N-1} \cos(\pi(n+1/2)k_1/N) \cos(\pi(n+1/2)k_2/N) \quad (3.24)$$

$$= \frac{w_{k_1}w_{k_2}}{N} \sum_{n=0}^{N-1} \cos(\pi(n+1/2)(k_1+k_2)/N) + \cos(\pi(n+1/2)(k_1-k_2)/N) \quad (3.25)$$

Il y a deux cas à étudier

1) $k_1 = k_2$.

Si $k_1 = k_2 = 0$, l'équation (3.25) sera égale à :

$$\langle \cos_N^{nk_1}, \cos_N^{nk_2} \rangle = \frac{1}{2N} \sum_{n=0}^{N-1} \cos(2\pi(n+1/2)(0)/N) + \sum_{n=0}^{N-1} 1 = \frac{2N}{2N} = 1 \quad (3.26)$$

Si $k_1 = k_2 \neq 0$, l'équation (3.25) sera égale à :

$$\langle \cos_N^{nk_1}, \cos_N^{nk_2} \rangle = \frac{1}{N} \sum_{n=0}^{N-1} \cos(2\pi(n+1/2)k_1/N) + \sum_{n=0}^{N-1} 1 = \frac{N}{N} = 1 \quad (3.27)$$

2) $k_1 \neq k_2$

$$\langle \cos_N^{nk_1}, \cos_N^{nk_2} \rangle = \frac{1}{N} \sum_{n=0}^{N-1} \cos(\pi(n+1/2)k_3/N) + \cos(\pi(n+1/2)k_4/N) = 0 \quad (3.28)$$

où

$$\begin{aligned} k_3 &= k_1 + k_2 \\ k_4 &= k_1 - k_2 \end{aligned} \quad (3.29)$$

3.6. Interprétation en algèbre linéaire :

La Transformée Discrète en Cosinus peut être considérée comme une transformation linéaire appliquée au vecteur $x[n]$ afin de donner le vecteur $X[k]$

$$X_c[k] = Dx[n] \quad (3.30)$$

Où D est une matrice de $N \times N$ avec les coefficients $d_{nk} = \cos(\pi(n+1/2)k/N)$

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,N-1} \\ d_{1,0} & d_{1,1} & \cdots & d_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ d_{N-1,0} & d_{N-1,1} & \cdots & d_{N-1,N-1} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (3.31)$$

3.7. Application de la TDC à une classe de signaux :

La Transformée Discrète en Cosinus et la transformée inverse sont programmées pour calculer et afficher les résultats de la classe des signaux du tableau (2.2).

Signal d'échelon : $x[n] = u[n]$

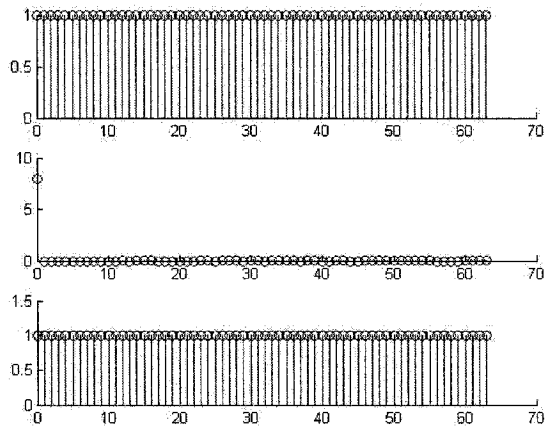


Figure 3. 1: Transformation d'un signal d'échelon par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal rectangulaire : $x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$

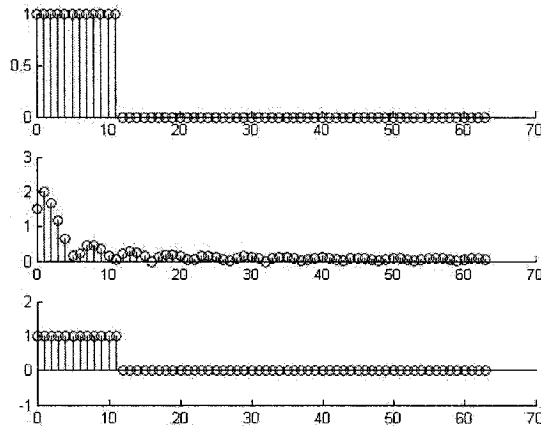


Figure 3. 2: Transformation d'un signal rectangulaire par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal sinusoïdal : $x[n] = \sin(2\pi fn)$

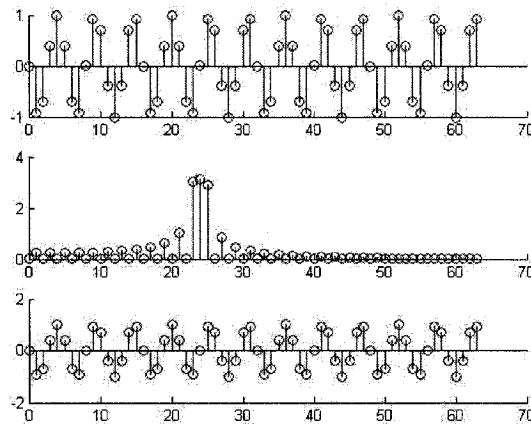


Figure 3. 3: Transformation d'un signal sinusoïdal par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal exponentiel : $x[n] = \exp(-\alpha n)$

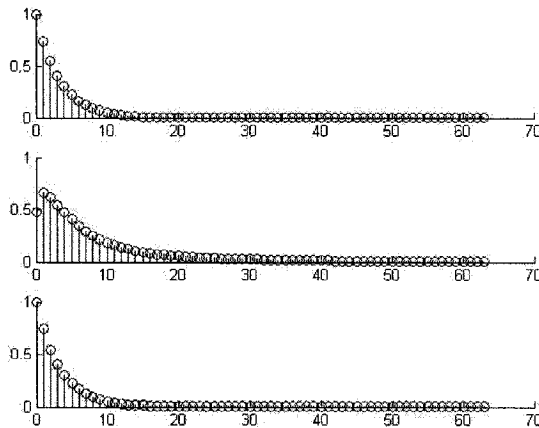


Figure 3. 4: Transformation d'un signal exponentiel par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal triangulaire: $x[n] = \Lambda_N$

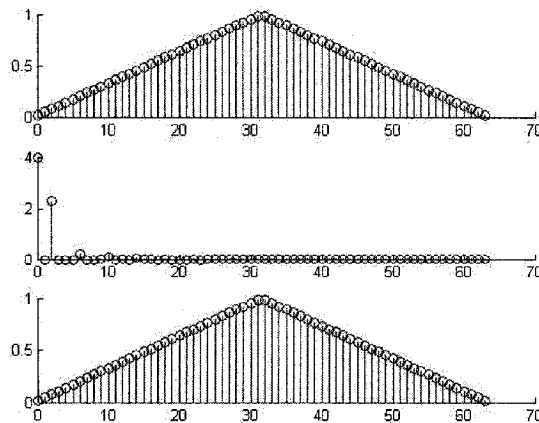


Figure 3. 5: Transformation d'un signal triangulaire par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal de rampe : $x[n] = n$

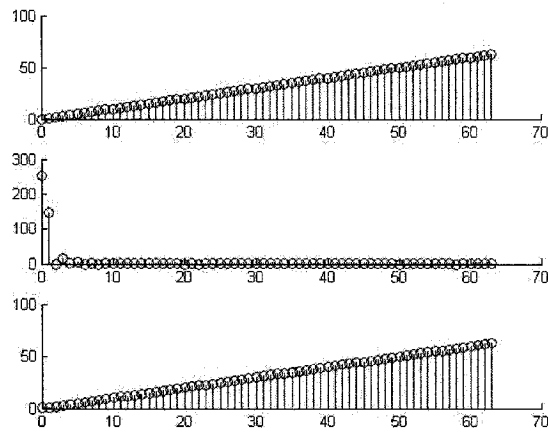


Figure 3. 6: Transformation d'un signal de rampe par la TDC.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal bidimensionnel : $x[n] = \text{image}$



Figure 3. 7: Résultat de la transformation d'une image par la TDC.

4. Transformée Discrète de Hartley (TDH)

4.1.Définition :

La Transformée Discrète de Hartley (TDH) est une transformée directement reliée à la Transformée Discrète de Fourier (TDF). Sa distinction principale de la TDF se situe au niveau des entrées/sorties. Elle transforme les entrées réelles en des sorties réelles, sans l'intervention d'aucun calcul de valeur complexe. La transformée continue a été introduite par R.V. Hartley en 1942 [7]. Ce n'est qu'à 1983 que la forme discrète est introduite par R. N. Bracewell [8, 10], qui a en plus, proposé une façon de factoriser sa matrice afin de produire un algorithme rapide. La Transformée Discrète de Hartley est donnée par l'équation suivante :

$$H_x[k] = \sum_{n=0}^{N-1} x[n] \text{cas}(2\pi nk / N) \quad (4.1)$$

où la fonction cas peut prendre une de ces formes suivantes :

$$\text{cas}(2\pi nk / N) = \cos(2\pi nk / N) + \sin(2\pi nk / N) \quad (4.2)$$

$$= \sqrt{2} \cos(2\pi nk / N - \pi / 4) \quad (4.3)$$

$$= \sqrt{2} \sin(2\pi nk / N + \pi / 4) \quad (4.4)$$

La Transformée Discrète de Hartley Inverse qui reproduit le signal $x[n]$ à partir de sa transformée $H_x[k]$, n'est que la transformée de Hartley elle-même avec une multiplication d'un facteur de $1/N$.

4.2.Algorithme Rapide pour la TDH :

Comme la Transformée Discrète de Fourier (TDF), le calcul direct de la Transformée Discrète de Hartley (TDH) d'une séquence de N points exige $O(N^2)$ opérations arithmétiques. En général, chaque algorithme rapide de la Transformée Discrète de Fourier mène à un algorithme correspondant pour la Transformée Discrète de Hartley. Par exemple, la Transformée Discrète de Hartley Rapide (TDHR) qui était introduite par Bracewell en 1984 [9] est l'analogue de l'algorithme de Cooley-Tukey de la Transformée Discrète de Fourier. Ces

algorithmes rapides réduisent la complexité du calcul pour obtenir les mêmes résultats avec seulement $O(N \log_2 N)$ opérations arithmétiques.

4.3. Transformée Discrète Bidimensionnelle de Hartley :

Comme la transformée de Fourier, la transformée de Hartley traite les signaux multi variants comme les images. Elle prend la forme suivante :

$$H_x[m_1, m_2] = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \text{cas}[2\pi(m_1 n_1 / N_1 + m_2 n_2 / N_2)] \quad (4.5)$$

et sa transformée inverse sera :

$$x[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} H_x[m_1, m_2] \text{cas}[2\pi(m_1 n_1 / N_1 + m_2 n_2 / N_2)] \quad (4.6)$$

4.4. Propriétés de la Transformée Discrète de Hartley :

Linéarité :

Si $x_1[n]$ et $x_2[n]$ sont deux signaux à durée limitée avec N_1 et N_2 , alors la TDH de

$$x[n] = ax_1[n] + bx_2[n] \quad (4.7)$$

est donnée par

$$H_x[k] = aH_{x_1}[k] + bH_{x_2}[k] \quad (4.8)$$

Changement d'échelle :

Si la Transformée Discrète de Hartley de $x[n]$ est $H_x[k]$, alors la TDH de

$$y[m] = x[an] \quad (4.9)$$

où a est une constante, sera :

$$H_y = \frac{1}{|a|} H_x[k/a] \quad (4.10)$$

Renversement temporel :

Si $x[n]$ est une séquence de N valeurs et sa Transformée Discrète de Hartley est $H_x[k]$. Alors la TDH de $x[-n]$ est donnée par

$$TDH\{x[-n]\} = H_x[-k] \quad (4.11)$$

Convolution :

Preuve :

La transformée de Hartley peut être déduite de celle de Fourier par l'équation suivante :

$$H_x[k] = \text{Re}\{X[k]\} - \text{Im}\{X[k]\} \quad (4.12)$$

et la transformée de Fourier de celle de Hartley par l'équation suivante :

$$X[k] = H_x^p[k] - jH_x^i[k] \quad (4.13)$$

où $H_x^p[k]$ et $H_x^i[k]$ sont les composantes paire et impaire de la transformée de Hartley respectivement. $H_x^p[k]$ et $H_x^i[k]$ sont calculés par les équations suivantes :

$$\begin{aligned} H_x^p[k] &= \frac{H_x[k] + H_x[-k]}{2} \\ H_x^i[k] &= \frac{H_x[k] - H_x[-k]}{2} \end{aligned} \quad (4.14)$$

D'après le théorème du convolution de la transformée de Fourier on a

$$\text{TDF}\{g[n] = x[n] \otimes y[n]\} = G[k] = X[k]Y[k] \quad (4.15)$$

Alors

$$H_g[k] = \text{Re}\{G[k]\} - \text{Im}\{G[k]\} \quad (4.16)$$

$$\begin{aligned} &= \frac{1}{2} \{ H_x[k]H_y[k] + H_x[-k]H_y[k] + \\ &\quad H_x[k]H_y[-k] - H_x[-k]H_y[-k] \} \end{aligned} \quad (4.17)$$

Cas exceptionnel :

1) si $x[n]$ est pair et $y[n]$ est impair ou vice-versa

$$x[n] \otimes y[n] = H_x[k]H_y[k] \quad (4.18)$$

2) si $x[n]$ est impair

$$x[n] \otimes y[n] = H_x[k]H_y[-k] \quad (4.19)$$

3) si $y[n]$ est impair

$$x[n] \otimes y[n] = H_x[-k]H_y[k] \quad (4.20)$$

4) si $x[n]$ et $y[n]$ sont impaires

$$x[n] \otimes y[n] = -H_x[k]H_y[k] \quad (4.21)$$

Les propriétés présentées ci-dessus se résument dans le tableau suivant :

Propriété	Fonction	Transformée Discrète de Hartley
	$x[n]$	$H_x[k]$
Linéarité	$x[n] = ax_1[n] + bx_2[n]$	$H_x[k] = aH_{x_1}[k] + bH_{x_2}[k]$
Renversement	$x[-n]$	$H_x[-k]$
Translation	$x[n - n_0]$	$\cos(2\pi kn_0 / N)H_x[k]$ $-\sin(2\pi kn_0 / N)H_x[-k]$
Modulation	$x[n]\cos(2\pi nk_0 / N)$	$\frac{1}{2}[H_x[k - k_0] + H_x[k + k_0]]$
Convolution	$x[n] \otimes y[n]$	$\frac{1}{2}\{H_x[k]H_y[k] + H_x[-k]H_y[k] +$ $H_x[k]H_y[-k] - H_x[-k]H_y[-k]\}$
Auto corrélation	$x[n] \odot x[n]$	$(H_x^p[k])^2 + (H_x^i[k])^2$
Multiplication	$x[n]y[n]$	$\frac{1}{2}\{H_x[k] \otimes H_y[k] + H_x[-k] \otimes H_y[k] +$ $H_x[k] \otimes H_y[-k] - H_x[-k] \otimes H_y[-k]\}$
Changement d'échelle	$x[an]$	$\frac{1}{ a }H_x[k/a]$
Dérivation	$\frac{dx[n]}{dn}$	$2\pi kH_x[-k]$

Tableau 4. 1: Propriétés de la Transformée Discrète de Hartley.

4.5.Orthogonalité de la Transformée Discrète de Hartley :

Pour que la transformée Discrète de Hartley (TDH) soit orthogonal, ces vecteurs de base doivent obéir à l'équation de l'orthogonalité (2.58).

Les vecteurs de base sont :

$$cas_N^{nk} = \cos(2\pi nk / N) = \sqrt{2} \cos(2\pi nk / N - \pi / 4) \quad (4.22)$$

En appliquant les vecteurs de base de cette transformée à l'équation (2.27), on aura

$$\langle \text{cas}_N^{nk_1}, \text{cas}_N^{nk_2} \rangle = 2 \sum_{n=0}^{N-1} \cos(2\pi nk_1 / N - \pi / 4) \cos(2\pi nk_2 / N - \pi / 4) \quad (4.23)$$

$$= \sum_{n=0}^{N-1} \cos(2\pi n(k_1 + k_2) / N - \pi / 2) + \cos(2\pi n(k_1 - k_2) / N) \quad (4.24)$$

Il y a deux cas à étudier

1) $k_1 = k_2$

L'équation (4.24) sera

$$\langle \text{cas}_N^{nk_1}, \text{cas}_N^{nk_2} \rangle = \sum_{n=0}^{N-1} \cos(4\pi nk_1 / N - \pi / 2) + \sum_{n=0}^{N-1} 1 = N \quad (4.25)$$

2) $k_1 \neq k_2$

L'équation (4.24) sera

$$\langle \text{cas}_N^{nk_1}, \text{cas}_N^{nk_2} \rangle = \sum_{n=0}^{N-1} \cos(2\pi n(k_1 + k_2) / N - \pi / 2) + \cos(2\pi n(k_1 - k_2)) \quad (4.26)$$

$$= \sum_{n=0}^{N-1} \cos(2\pi n(k_3) / N - \pi / 2) + \cos(2\pi n(k_4)) = 0 \quad (4.27)$$

où

$$\begin{aligned} k_3 &= k_1 + k_2 \\ k_4 &= k_1 - k_2 \end{aligned} \quad (4.28)$$

4.6. Interprétation en algèbre linéaire :

La Transformée Discrète de Hartley peut être considérée comme une transformation linéaire appliquée sur un vecteur $x[n]$ afin de donner le vecteur $H_x[k]$

$$H_x[k] = Hx[n] \quad (4.29)$$

Où H est une matrice avec les coefficients $h_{nk} = \text{cas}(2\pi nk / N)$

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,N-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ h_{N-1,0} & h_{N-1,1} & \cdots & h_{N-1,N-1} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (4.30)$$

4.7. Application de la TDH à une classe des signaux :

La Transformée Discrète de Hartley et sa transformée inverse sont programmées pour calculer et afficher les résultats de la classe des signaux du tableau (2.2).

Signal d'échelon : $x[n] = u[n]$

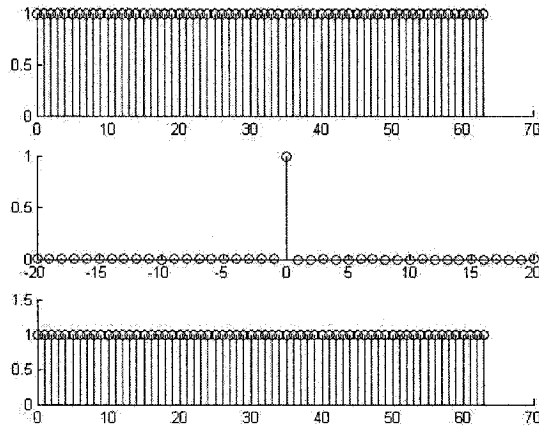


Figure 4. 1: Transformation d'un signal d'échelon par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.

Le troisième le signal reconstruit.

Signal rectangulaire : $x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$

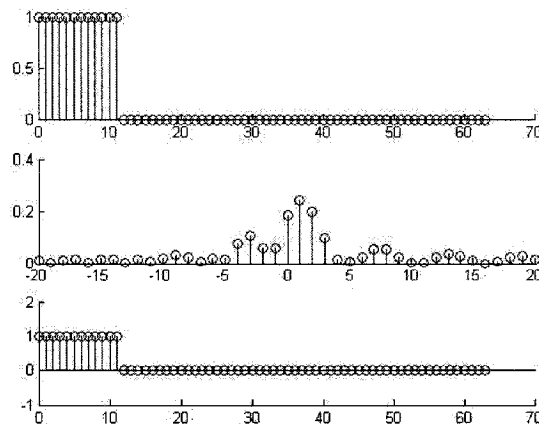


Figure 4. 2: Transformation d'un signal rectangulaire par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.

Le troisième le signal reconstruit.

Signal sinusoïdal : $x[n] = \sin(2\pi fn)$

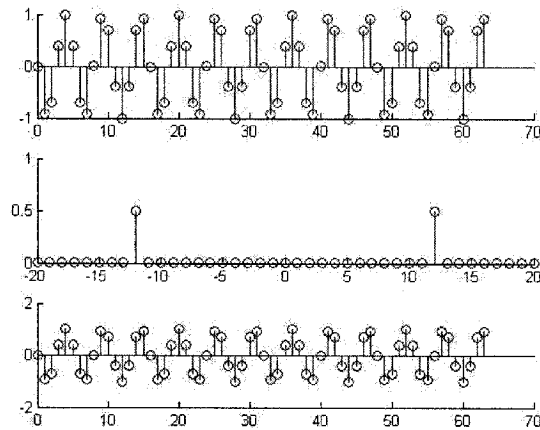


Figure 4.3: Transformation d'un signal sinusoïdal par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal exponentiel : $x[n] = \exp(-\alpha n)$

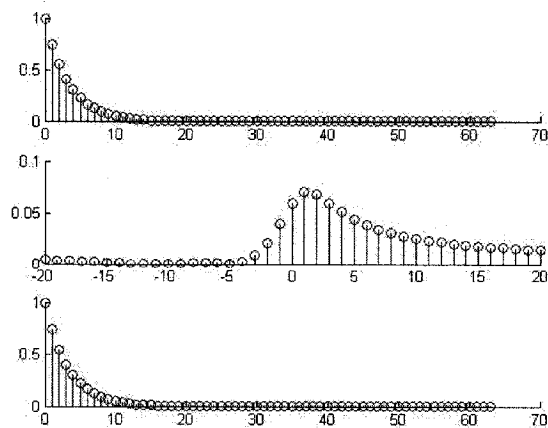


Figure 4.4: Transformation d'un signal exponentiel par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal triangulaire : $x[n] = \Lambda_N$

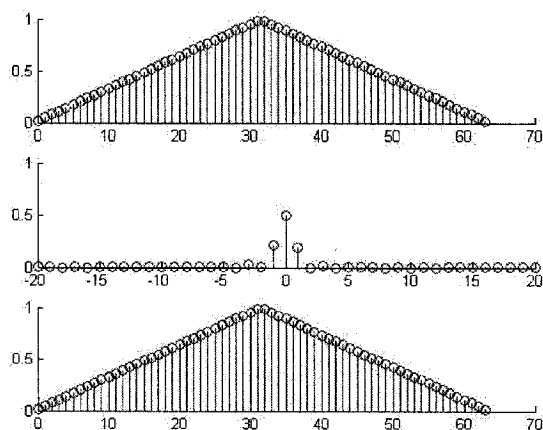


Figure 4.5: Transformation d'un signal triangulaire par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal de rampe : $x[n] = n$

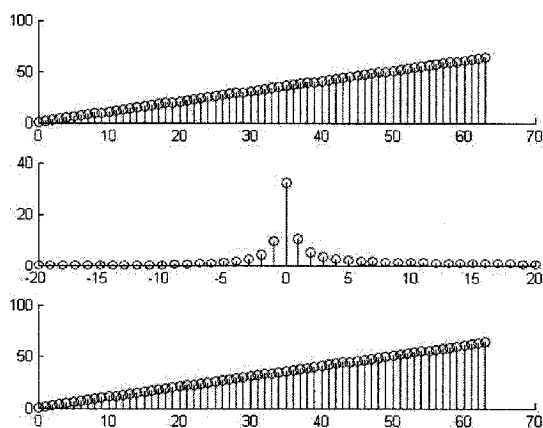


Figure 4.6: Transformation d'un signal de rampe par la TDH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal bidimensionnel : $x[n] = \text{image}$

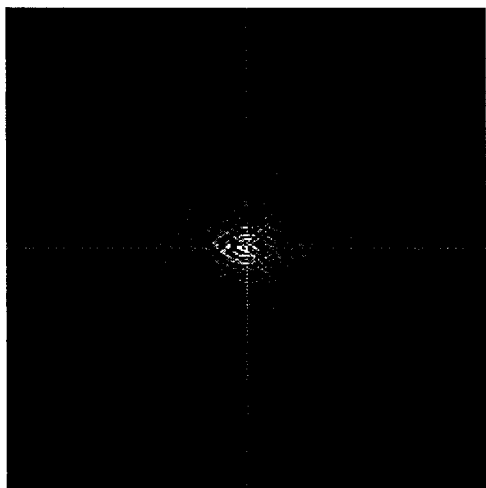


Figure 4. 7: Résultat de la transformation d'une image par la TDH.

5. Transformées Discrète de Walsh Hadamard (TDWH)

Dans ce chapitre on traitera la transformée de Walsh Hadamard qui est la plus reconnue parmi la famille des transformées orthogonales non sinusoïdales. Cette transformée a réussi à être utilisée dans plusieurs applications de traitement numérique des signaux à cause de son simple calcul qui exige seulement des additions et des soustractions.

5.1.Définition :

La Transformée de Walsh Hadamard prend différentes formes suivant comment les lignes de la matrice sont arrangées. On note ici trois formes : l'ordre naturel, l'ordre dyadique et l'ordre de séquence. Pour chaque ordre on fournira deux définitions, une matricielle et l'autre exponentielle.

5.1.1 Transformée de Walsh Hadamard ordre naturel :

L'ordre naturel est aussi connu sous l'ordre de Hadamard et encore sous la Transformée de BIFORE (BInary FOurier REpresentation) [23, 24].

La matrice de cette transformée pour $N=2$ est donnée par l'équation suivante :

$$H(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5.1)$$

La génération de la matrice de Walsh Hadamard pour $N > 2$ dans l'ordre naturel est faite par une opération récursive comme l'équation suivante le montre :

$$H_{nat}(k) = \begin{bmatrix} H_{nat}(k-1) & H_{nat}(k-1) \\ H_{nat}(k-1) & -H_{nat}(k-1) \end{bmatrix} \quad (5.2)$$

où

$$\begin{aligned} k &= 1, 2, 3, \dots, n \\ n &= \log_2 N \end{aligned} \quad (5.3)$$

Exemple pour $N=8$:

$$H_{nat}(2) = \begin{bmatrix} H_{nat}(1) & H_{nat}(1) \\ H_{nat}(1) & -H_{nat}(1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (5.4)$$

$$H_{nat}(3) = \begin{bmatrix} H_{nat}(2) & H_{nat}(2) \\ H_{nat}(2) & -H_{nat}(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{matrix} \quad (5.5)$$

où le vecteur à droite de la matrice, équation (5.5), indique le nombre de changement de signe dans chaque ligne.

La transformée inverse de la transformée dans l'ordre naturel a la même matrice de celle du sens direct avec une multiplication par un facteur de $1/N$.

5.1.1.1 Définition exponentielle de l'ordre naturel :

[25] a donné une définition exponentielle à la transformée de Walsh Hadamard de l'ordre naturel :

$$B_x[k] = \sum_{n=0}^{N-1} x[n](-1)^{\langle n, k \rangle_N} \quad (5.6)$$

où $B_x[k]$ est la transformée d'ordre naturel de Walsh Hadamard et la fonction $\langle n, k \rangle$ est définie par l'équation suivant :

$$\langle n, k \rangle_N = \sum_{i=0}^{\log_2 N - 1} k_i n_i \quad (5.7)$$

Les termes k_i et n_i dans l'équation (5.7) sont les coefficients de la représentation binaire de k et n respectivement.

La transformée inverse dans la forme exponentielle est :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} B_x[k](-1)^{\langle n, k \rangle_N} \quad (5.8)$$

5.1.2 Transformée de Walsh Hadamard ordre dyadique :

L'ordre dyadique est produit par une multiplication antérieure de la matrice d'ordre naturel de Hadamard par la matrice de permutation de bit inversé [27]. Cet ordre est introduit par Paley [26] et est parfois désigné par son nom.

Exemple de la matrice d'ordre dyadique pour N=8 :

La figure (5.1) au-dessous montre la transformation de l'ordre normal à l'ordre bit inversé pour une séquence binaire de 8 points.

<i>order normal</i>	<i>ordre bit – inversé</i>
000	000
001	100
010	010
011	110
100	001
101	101
110	011
111	111

Figure 5. 1: Conversion d'ordre normal à ordre bit-inversé.

Après avoir obtenu l'ordre bit inversé, on le multiplie par la matrice d'identité pour générer la matrice de permutation, équation (5.9).

$$R_8 = \text{diag}[0, 4, 2, 6, 1, 5, 3, 7].I_8 = \begin{bmatrix} 1 & & & & & & & \\ & & & 1 & & & & \\ & & 1 & & & & & \\ & & & & & & 1 & \\ & 1 & & & & & & \\ & & & & & 1 & & \\ & & & & 1 & & & \\ & & & & & & & 1 \end{bmatrix} \quad (5.9)$$

L'ordre dyadique peut donc être déduit maintenant par la multiplication de (5.9) par $H_{nat}(3)$.

$$H_{dyad}(3) = R_8 \cdot H_{nat}(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 3 \\ 2 \\ 7 \\ 6 \\ 4 \\ 5 \end{matrix} \quad (5.10)$$

La matrice de la transformée inverse de l'ordre dyadique est la même matrice de sens directe multipliée par un facteur de $1/N$.

5.1.2.1 Définition exponentielle de l'ordre dyadique :

La définition exponentielle de l'ordre dyadique est donnée par l'équation suivante [25] :

$$P_x[k] = \sum_{n=0}^{N-1} x[n] (-1)^{\langle p(n), k \rangle_N} \quad (5.11)$$

où $P_x[k]$ est la transformée d'ordre dyadique de Walsh Hadamard et la fonction $\langle p(n), k \rangle$ est définie par l'équation suivante :

$$\langle p(n), k \rangle_N = \sum_{i=0}^{\log_2 N - 1} n_{\log_2 N - 1 - i} k_i \quad (5.12)$$

Les termes k_i et n_i dans l'équation (5.12) sont les coefficients de la représentation binaire de k et n respectivement.

La transformée inverse de la forme exponentielle est :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} P_x[k] (-1)^{\langle p(n), k \rangle_N} \quad (5.13)$$

5.1.3 Transformée de Walsh Hadamard ordre de séquence :

La transformée de Walsh Hadamard d'ordre de séquence est encore connue sous le nom d'« ordre de Walsh ». Pour générer la matrice de cette transformée, il faut multiplier la matrice d'ordre dyadique par une matrice de permutation obtenue par la conversion du code binaire au code de gray [26].

Pour faire cette conversion, on applique l'équation suivante :

$$g_i = b_{i+1} \otimes b_i \quad (5.14)$$

où les termes b_i et g_i sont les coefficients de la représentation binaire et de gray respectivement.

Exemple de la matrice d'ordre séquence pour N=8 :

La figure (5.2) au-dessous montre la transformation d'une séquence binaire de 8 points à une séquence de gray de même nombre de points par l'application de l'équation (5.14).

<i>ordre binaire</i>	<i>ordre gray</i>
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Figure 5. 2: Conversion d'ordre binaire à ordre de gray.

Par la même procédure de l'ordre dyadique on multiplie l'ordre obtenu par la conversion du code binaire au code de gray par la matrice d'identité. Le résultat de cette multiplication nous donne la matrice de permutation, équation (5.15).

$$G_8 = \text{diag}[0,1,3,2,6,7,5,4].I_8 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \quad (5.15)$$

Finalement la matrice d'ordre de séquence de Walsh Hadamard pour $N=8$ est obtenue après avoir multiplié la matrice obtenue par l'équation (5.15) et par la matrice d'ordre dyadique.

$$H_{seq}(3) = G_8 \cdot H_{dyad}(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \quad (5.16)$$

5.1.3.1 Définition exponentielle de l'ordre de séquence :

La définition exponentielle de la Transformée de Walsh Hadamard de l'ordre de séquence est donnée par l'équation suivante [25] :

$$W_x[k] = \sum_{n=0}^{N-1} x[n] (-1)^{\langle n, r(k) \rangle_N} \quad (5.17)$$

où $W_x[k]$ est la transformée d'ordre de séquence de Walsh Hadamard et la fonction $\langle n, r(k) \rangle$ est définie par :

$$\langle n, r(k) \rangle_N = \sum_{i=0}^{\log_2 N - 1} r_i(k) n_i \quad (5.18)$$

où

$$r_0(k) = k_{m-1}, \quad r_1(k) = k_{m-1} + k_{m-2}, \quad \dots, \quad r_{m-1} = k_1 + k_0 \quad (5.19)$$

Les termes k_i et n_i dans les équations (5.18) et (5.19) sont les coefficients de la représentation binaire de k et n respectivement.

La transformée inverse dans la forme exponentielle est :

$$W_x[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] (-1)^{\langle n, r(k) \rangle_N} \quad (5.20)$$

5.2.Algorithmes Rapides pour la TDWH :

Pour le calcul rapide de cette transformée dans ses différents ordres, des algorithmes ont été proposés qui ressemblent à celle de Cooley-Tukey pour la Transformée Discrète de Fourier. Autre approche peut être trouvée dans la littérature pour le calcul rapide de cette transformée, parmi ces approches celle de [27]. Dans cette approche, une factorisation de la matrice de la Transformée de Walsh Hadamard dans son ordre naturel qui ressemble à l'algorithme avec permutation antérieur (Post-permutation) de la Transformée de Fourier fut proposée. Pour le calcul de l'ordre dyadique l'algorithme proposé correspond à celle de l'algorithme Entrée-Ordonnée Sortie-Ordonnée (Ordered-input Ordered Output) de la Transformée de Fourier. La matrice de Walsh Hadamard dans l'ordre de séquence est factorisée d'une façon pareille pour générer un algorithme rapide.

Ordre naturel:

La transformée de Hadamard pour cet ordre est obtenue par l'application successive de la multiplication de Kronecker de la matrice de base H_2 . Alors

$$H_{N,nat} = H_{N/2,nat} \times H_2 = H_{N/4,nat} \times H_2 \times H_2 = \dots \quad (5.21)$$

On note que

$$\begin{aligned} H_N &= \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} H_{N/2} & \\ & H_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\ &= (H_{N/2} \times I_2)(I_{N/2} \times H_2) \end{aligned} \quad (5.22)$$

Si on continue la décomposition de la matrice de même façon jusqu'à H_N dépend seulement sur la matrice de base H_2 et si on applique la propriété donnée par l'équation (2.10), H_N sera égale à

$$\begin{aligned} H_N &= (H_2 \times I_{N/2}) \\ &\quad \cdot (I_2 \times H_2 \times I_{N/4}) \dots (I_{N/2k} \times H_2 \times I_k) \\ &\quad \dots (I_{N/4} \times H_2 \times I_2)(I_{N/2} \times H_2) \end{aligned} \quad (5.23)$$

Cette équation peut être écrite d'une façon compacte

$$H_N = \prod_{i=1,2,\dots}^n (I_{2^{(i-1)}} \times H_2 \times I_{2^{(n-i)}}) \quad (5.24)$$

La permutation de mélange parfaite possède la propriété suivante :

$$P_N^{-(n-i)} (I_{N/2} \times H_2) P_N^{(n-i)} = I_{2^{(i-1)}} \times H_2 \times I_{2^{(n-i)}} \quad (5.25)$$

En remplaçant l'équation (5.25) dans (5.24) et en posant $C = (I_{N/2} \times H_2)$ on obtient

$$H_N = \prod_{i=1,2,\dots}^n P_N^{-(n-i)} C P_N^{(n-i)} \quad (5.26)$$

Utilisant la propriété suivante

$$\prod_{i=1,2,\dots}^n A_i B_i = A_1 \left[\prod_{i=2,3,\dots}^n B_{(i-1)} A_i \right] B_n \quad (5.27)$$

L'équation (5.26) sera égale à

$$H_N = \prod_{i=1,2,\dots}^n P_N C \quad (5.28)$$

Ordre Dyadique :

C'était mentionné avant, que la matrice d'ordre dyadique est obtenue de celle de l'ordre naturel par une multiplication antérieure par une matrice de permutation qui performe l'ordonnance de bit inverse. La matrice de permutation de dimension de N points qui performe l'ordonnance de bit inverse peut être écrite sous la forme d'un produit de $n = \log_2 N$ matrices de petite dimension en utilisant la matrice de mélange parfaite

$$K_N = \prod_{i=1,2,\dots}^n P_{2^{(n-i+1)}} \times I_{2^{(i-1)}} \quad (5.29)$$

En appliquant cette matrice à la matrice d'ordre naturel on obtient

$$H_{N,dyd} = \prod_{i=1,2,\dots}^n P_{2^{(n-i+1)}} \times I_{2^{(i-1)}} \cdot \prod_{i=1,2,\dots}^n (I_{2^{(i-1)}} \times H_2 \times I_{2^{(n-i)}}) \quad (5.30)$$

En appliquant les propriétés suivantes sur la dernière équation

$$P_k (A_{k/2} \times I_2) P_k^{-1} = I_2 \times A_{k/2} \quad (5.31)$$

et

$$P_k (ABC\dots) P_k^{-1} = P_k A P_k^{-1} P_k B P_k^{-1} P_k C P_k^{-1} \dots \quad (5.32)$$

On obtient

$$H_{N,dyd} = \prod_{i=1,2,\dots}^n I_{2^{(n-i)}} \times [P_{2^i} (I_{2^{(i-1)}} \times H_2)] = \prod_{i=1,2,\dots}^n (I_{2^{(n-i)}} \times P_{2^i}) (I_{2^{(n-i)}} \times H_2) \quad (5.33)$$

Et en remplaçant $C = (I_{N/2} \times H_2)$, l'équation (5.33) sera égale à :

$$H_{N,dyd} = \prod_{i=1,2,\dots}^n (I_{2^{(n-i)}} \times P_{2^i}) C \quad (5.34)$$

Ordre de séquence :

La factorisation de la matrice de cet ordre ne correspond pas à aucun autre algorithme comme les deux premiers ordres qui sont présentés ci-dessus.

La matrice de cet ordre peut être écrite

$$\begin{aligned} H_{N,seq} &= P_N P_N^{-1} H_{N,seq} \\ &= P_N H_N' \end{aligned} \quad (5.35)$$

La matrice H_N' a la possibilité d'être factorisée et sur cela que cet algorithme est basée. Après la factorisation de H_N' l'équation (5.35) sera égale à

$$H_{N,seq} = P_N \begin{bmatrix} H_{N/2,seq} & D_{N/2} H_{N/2,seq} \\ H_{N/2,seq} & -D_{N/2} H_{N/2,seq} \end{bmatrix} \quad (5.36)$$

où $D_{N/2}$ est une matrice diagonale dont ses éléments changent de +1 à -1 alternativement.

L'équation (5.36) peut être écrite d'une façon compacte

$$H_{N,seq} = P_N (I_{N/2} \times H_2) D_N' (H_{N/2} \times I_2) \quad (5.37)$$

où D_N' est une matrice quasi-diagonale égale à

$$D_N' = \begin{bmatrix} I_{N/2} & \\ & D_{N/2} \end{bmatrix} \quad (5.38)$$

En répétant la factorisation de $H_{N,seq}$ jusqu'à elle ne dépend que de H_2 et en appliquant la propriété donnée par l'équation (2.10), l'équation (5.37) sera égale à

$$\begin{aligned}
H_{N,seq} = & P_N (I_{N/2} \times H_2) D_N' (P_{N/2} \times I_2) (I_{N/4} \times H_2 \times I_2) \\
& \cdot (D_{N/2}' \times I_2) \dots (P_{N/k} \times I_k) (I_{N/2k} \times H_2 \times I_k) \\
& \cdot (D_{N/k}' \times I_k) \dots (P_{N/8} \times I_8) (I_4 \times H_2 \times I_{N/8}) \\
& \cdot (D_8' \times I_{N/8}) (P_4 \times I_{N/4}) (I_2 \times H_2 \times I_{N/4}) \\
& \cdot (D_4' \times I_{N/4}) (H_2 \times I_{N/2})
\end{aligned} \tag{5.39}$$

Cette équation peut être réécrite d'une façon compacte

$$H_{N,seq} = \prod_{i=1,2,\dots}^n (P_{2^{(n-i-1)}} \times I_{2^{(i-1)}}) (I_{2^{(n-i)}} \times H_2 \times I_{2^{(i-1)}}) (D_{2^{(n-i-1)}}' \times I_{2^{(i-1)}}) \tag{5.40}$$

En appliquant la propriété de la matrice de permutation donnée par l'équation (5.25) sur cette dernière équation on obtient

$$H_{N,seq} = \prod_{i=1,2,\dots}^n (P_{2^{(n-i-1)}} \times I_{2^{(i-1)}}) P_{2^n}^{-(i-1)} C P_{2^n}^{(i-1)} (D_{2^{(n-i-1)}}' \times I_{2^{(i-1)}}) \tag{5.41}$$

où

$$C = I_{N/2} \times H_2 \tag{5.42}$$

Finalement, en utilisant les équations (5.27), (5.31) et (5.32), l'équation (5.41) sera égale à

$$H_{N,seq} = P_{2^n} \left\{ \prod_{i=1,2,\dots}^n P_{2^n}^{-1} (I_{2^{(i-1)}} \times P_{2^{(n-i+1)}}) C (I_{2^{(i-1)}} \times D_{2^{(n-i+1)}}') \right\} P_{2^n}^{-1} \tag{5.43}$$

5.3. Transformée de Walsh Généralisée et Algorithmes Rapides :

5.3.1 Transformée de Walsh Généralisée :

La Transformée de Walsh Généralisée (TWG) [20], fournit des matrices pour tous les ordres pour n'importe quelle base. On supposons que la séquence numérique à transformée peut prendre la forme ci-dessous

$$N = p^n \tag{5.44}$$

où p est la base. La matrice de base de la transformée sera égale à

$$W_p = \frac{1}{\sqrt{p}} \begin{bmatrix} w^0 & w^0 & \dots & w^0 \\ w^0 & w^1 & \dots & w^{p-1} \\ \cdot & \cdot & \dots & \cdot \\ w^0 & w^{p-1} & \dots & w^1 \end{bmatrix} \quad (5.45)$$

où

$$w = \exp(j2\pi / p) \quad (5.46)$$

Ordre naturel généralisé:

La séquence qui était égale au nombre de changement de signe dans les lignes de la matrice pour la base 2, devient la somme des distances entre valeurs propres successives sur une ligne divisée par (p-1). La distance entre w^s et w^r est:

$$\begin{aligned} s - r \text{ si } s \geq r \\ p + (s - r) \text{ si } s < r \end{aligned} \quad (5.47)$$

La génération de la matrice de la Transformée de Walsh Hadamard Généralisée dans l'ordre naturel est faite par le produit Kronecker comme l'équation suivant montre :

$$W_{p^2} = \frac{1}{p} W_p \times W_p \quad (5.48)$$

Walsh-Paley généralisée (p-adique):

L'ordre p-adique est produit par la multiplication de la matrice d'ordre naturel par une matrice de permutation. La matrice de permutation $R_N^{(p)}$ est obtenue par la conversion de la séquence de base p en une séquence à bit inverse. Si $R_N^{(p)}$ est appliquée sur un vecteur x de dimension $N = p^n$ donne un vecteur y égale à:

$$\begin{aligned} y_{k_{n-1}k_{n-2}\dots k_0} = x_{k_0k_1\dots k_{n-2}k_{n-1}}; \quad k_i \in \{0,1,\dots,p-1\} \\ i = 0,1,\dots,n-1 \end{aligned} \quad (5.49)$$

alors la matrice p-adique sera égale à

$$W_{p\text{-adique}} = R_N^{(p)} W_{p,\text{nat}} \quad (5.50)$$

Walsh-Kaczmarz généralisée (ordre de séquence):

La génération de cet ordre dans la forme généralisée est effectuée par la multiplication de la matrice p-adique par la matrice de permutation obtenue par la

conversion du code de base p au code de gray. La matrice de permutation $G_N^{(p)}$ produit un vecteur y d'un autre x comme c'est indique ci-dessus:

$$y_{k_{n-1}k_{n-2}\dots k_0} = x_{a_{n-1}a_{n-2}\dots a_0} \quad (5.51)$$

où

$$\begin{aligned} k_i \oplus a_{i+1} &= a_i; \quad k_i, a_i \in \{0, 1, \dots, p-1\} \\ i &= 0, 1, \dots, n-1 \end{aligned} \quad (5.52)$$

alors la matrice d'ordre de séquence généralisé peut être écrite comme le suivante

$$W_{seq} = G_N^{(p)} W_{p\text{-adique}} \quad (5.53)$$

5.3.2 Factorisation pour un algorithme optimal :

Corinthios [20] a proposé des factorisations pour des processeurs rapides de forme p^2 – *optimal* . La factorization de ces matrices est faite dans leurs formes généralisées.

Ordre naturel :

La factorisation de la matrice de la Transformée de Walsh Généralisée d'ordre naturel est donnée par l'équation suivante [20] :

$$W_{N,WN} = \prod_{i=1,2,\dots}^n P_N C_N = \prod_{i=1,2,\dots}^n P_N (I_{N/p} \times W_p) \quad (5.54)$$

En réécrivant la dernière équation comme ci-dessous

$$W_{N,WN} = P \left\{ \prod_{i=0,1,2,\dots}^{n-1} C P_N \right\} P^{-1} = P \left\{ \prod_{i=0,1,2,\dots}^{n-1} F \right\} P^{-1} \quad (5.55)$$

On remarque que la matrice $F = CP$ est p^2 – *optimal* . Cette factorisation mène à une structure optimale avec un nombre minimal de partitions de la mémoire dans un algorithme sans mélange.

Ordre p-adique :

La factorisation de la matrice de la Transformée de Walsh-Paley Généralisée (p-adique) est donnée par l'équation suivante [20] :

$$W_{N,WP} = \prod_{i=0,1,2,\dots}^{n-1} J_i C_N \quad (5.56)$$

où

$$J_i = (I_{p^{n-i-1}} \times P_{p^{i+1}}) = H_{n-i-1} \quad (5.57)$$

Pour changer le produit $J_i C_N$ à $C_N J_{i+1}$ on introduit un nouveau opérateur Q_i égale à

$$\begin{aligned} Q_i &= C_N J_{i+1} = C_N H_{n-i-2}; \quad i = 0, 1, \dots, n-2 \\ Q_{n-1} &= C_N \end{aligned} \quad (5.58)$$

En remplaçant Q_i dans l'équation (5.56) on obtient

$$W_{N,WP} = \prod_{i=0,1,2,\dots}^{n-1} Q_i \quad (5.59)$$

Ordre de séquence :

La factorisation de la matrice de la Transformée de Walsh-Kaczmarz Généralisée (ordre de séquence) est donnée par l'équation suivante [20] :

$$W_{N,WK} = P \left\{ \prod_{i=0,1,2,\dots}^{n-1} P^{-1} H_i C_N E_i \right\} P^{-1} \quad (5.60)$$

où

$$\begin{aligned} H_i &= I_{p^i} \times P_{p^{n-i}} \\ E_i &= I_{p^i} \times D_{p^{n-i}} \end{aligned} \quad (5.61)$$

Supposons que $G_i = C_N E_i$ alors l'équation (5.60) sera égale à

$$W_{N,WK} = P \left\{ \prod_{i=0,1,2,\dots}^{n-1} P^{-1} H_i G_i \right\} P^{-1} \quad (5.62)$$

Cette équation peut être réécrite dans sa forme finale

$$W_{N,WK} = P \left\{ \prod_{i=0,1,2,\dots}^{n-1} \Gamma_i \right\} P^{-1} \quad (5.63)$$

où

$$\begin{aligned} \Gamma_i &= P^{-1} G_i S_{i+1}; \quad i = 1, 2, \dots, n-1 \\ \Gamma_0 &= G_0 S_1 \end{aligned} \quad (5.64)$$

et avec

$$S_i = P^{-1} H_i P \quad (5.65)$$

Dans [59], l'auteur a introduite une formalisation et un algorithme pour configurer des processeurs avec un parallélisme massif pour les Transformées Généralisées.

5.4. Transformée Discrète Bidimensionnelle de Walsh Hadamard :

La matrice de la Transformée de Walsh sous toutes les formes est orthogonale et symétrique alors la transformée bidimensionnelle peut être définie en utilisant l'algorithme de ligne colonne [25]. Cet algorithme stipule que la matrice unidimensionnelle appliqué sur toutes les lignes du signal bidimensionnelle suivie par une notre application de la même matrice sur les colonnes, produit la transformée bidimensionnelle du signal.

$$X[k, l] = H_{nat} x H_{nat}^{-1} \quad (5.66)$$

$$X[k, l] = H_{dyad} x H_{dyad}^{-1} \quad (5.67)$$

$$X[k, l] = H_{seq} x H_{seq}^{-1} \quad (5.68)$$

La transformée inverse bidimensionnelle peut être obtenue par une application dans le sens contraire de l'algorithme de ligne colonne : la matrice unidimensionnelle est appliquée d'abord sur toutes les colonnes du signal bidimensionnel et ensuite sur toutes les lignes.

$$x = H_{nat}^{-1} X[k, l] H_{nat} \quad (5.69)$$

$$x = H_{dyad}^{-1} X[k, l] H_{dyad} \quad (5.70)$$

$$x = H_{seq}^{-1} X[k, l] H_{seq} \quad (5.71)$$

Il y a une autre définition de la transformée bidimensionnelle pour tous les ordres donnée par [19] dans la forme exponentielle.

$$B[k_1, k_2] = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x[n_1, n_2] (-1)^{\langle n_1, k_1 \rangle_{N_1} + \langle n_2, k_2 \rangle_{N_2}} \quad (5.72)$$

$$P[k_1, k_2] = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x[n_1, n_2] (-1)^{\langle p(n_1), k_1 \rangle_{N_1} + \langle p(n_2), k_2 \rangle_{N_2}} \quad (5.73)$$

$$W[k_1, k_2] = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x[n_1, n_2] (-1)^{\langle n_1, r(k_1) \rangle_{N_1} + \langle n_2, r(k_2) \rangle_{N_2}} \quad (5.74)$$

et les transformées inverses sont :

$$x[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} B[k_1, k_2] (-1)^{\langle n_1, k_1 \rangle_{N_1} + \langle n_2, k_2 \rangle_{N_2}} \quad (5.75)$$

$$x[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} P[k_1, k_2] (-1)^{\langle p(n_1), k_1 \rangle_{N_1} + \langle p(n_2), k_2 \rangle_{N_2}} \quad (5.76)$$

$$x[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} W[k_1, k_2] (-1)^{\langle n_1, r(k_1) \rangle_{N_1} + \langle n_2, r(k_2) \rangle_{N_2}} \quad (5.77)$$

5.5. Propriétés de la Transformée Discrète de Walsh Hadamard :

Linéarité :

Si $x_1[n]$ et $x_2[n]$ sont deux signaux à durée limitée avec N_1 et N_2 , alors la WHT de

$$x[n] = ax_1[n] + bx_2[n] \quad (5.78)$$

est donnée par

$$B[k] = aB_1[k] + bB_2[k] \quad (5.79)$$

Convolution et corrélation dyadique :

Si $x[n]$ et $h[n]$ sont deux signaux réels tel que

$$WHT \{x[n]\} = B_x[k] \quad (5.80)$$

$$WHT \{h[n]\} = B_h[k] \quad (5.81)$$

et leur convolution dyadique est donnée par

$$y[n] = \frac{1}{N} \sum_{u=0}^{N-1} x[u] h[n \oplus u] \quad (5.82)$$

où \oplus indique l'opération de soustraction de modulo de 2. Alors

$$B_y[k] = B_x[k] B_h[k] \quad (5.83)$$

et leur corrélation est donnée par

$$r_{xh}[n] = \frac{1}{N} \sum_{u=0}^{N-1} x[u] h[n \bullet u] \quad (5.84)$$

où $[n \bullet u]$ indique l'opération de soustraction de modulo de 2. Alors

$$B_{xh}[k] = B_x[k] B_h[k] \quad (5.85)$$

5.6. Orthogonalité de la Transformée de Walsh Hadamard :

Pour que la transformée de Walsh Hadamard soit orthogonale, on doit démontrer que la fonction de base de chaque ordre obéit à l'équation (2.58).

La fonction de base de cette transformée dans l'ordre naturel est donnée dans l'équation (5.6). Alors

$$\begin{aligned} \langle (-1)^{\langle n, k_1 \rangle}, (-1)^{\langle n, k_2 \rangle} \rangle &= \sum_{n=0}^{N-1} (-1)^{\langle n, k_1 \rangle} (-1)^{\langle n, k_2 \rangle} \\ &= \sum_{n=0}^{N-1} (-1)^{\langle n, k_1 \rangle + \langle n, k_2 \rangle} \end{aligned} \quad (5.86)$$

Il y a deux cas à étudier :

$$1) k_1 = k_2$$

$$\langle (-1)^{\langle n, k_1 \rangle}, (-1)^{\langle n, k_2 \rangle} \rangle = \sum_{n=0}^{N-1} (-1)^{2\langle n, k_1 \rangle} = \sum_{n=0}^{N-1} 1 = N \quad (5.87)$$

$$2) k_1 \neq k_2$$

$$\langle (-1)^{\langle n, k_1 \rangle}, (-1)^{\langle n, k_2 \rangle} \rangle = \sum_{n=0}^{N-1} (-1)^{\langle n, k_1 \rangle + \langle n, k_2 \rangle} = 0 \quad (5.88)$$

La démonstration au-dessus est faite pour l'ordre naturel. Comme les matrices de l'ordre dyadique et de séquences sont des permutations sur les lignes de la matrice de l'ordre naturel, alors l'orthogonalité est valide encore pour eux.

5.7. Application de la TDWH (ordre naturel) à une classe des signaux :

La Transformée Discrète de Walsh Hadamard (ordre naturel) et la transformée inverse sont programmées pour calculer et afficher les résultats de la même classe des signaux du tableau (2.2)

Signal d'échelon : $x[n] = u[n]$

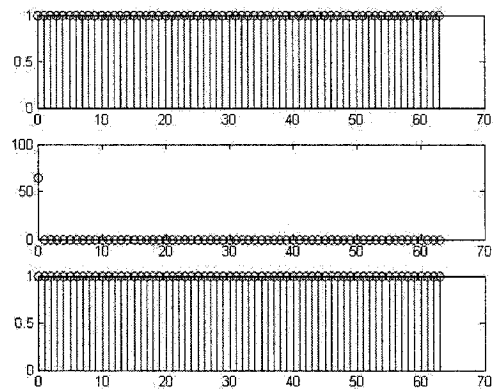


Figure 5.3: Transformation d'un signal d'échelon par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal rectangulaire : $x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$

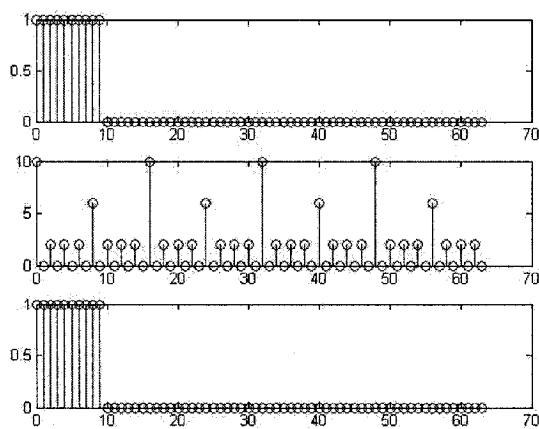


Figure 5.4: Transformation d'un signal rectangulaire par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal sinusoïdal : $x[n] = \sin(2\pi fn)$

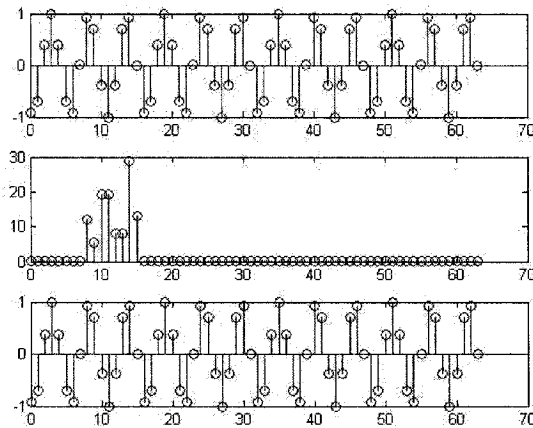


Figure 5.5: Transformation d'un signal sinusoïdal par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal exponentiel : $x[n] = \exp(-\alpha n)$

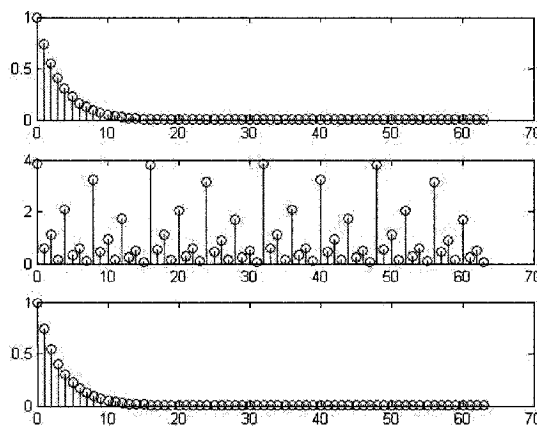


Figure 5.6: Transformation d'un signal exponentiel par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal triangulaire : $x[n] = \Lambda_N$

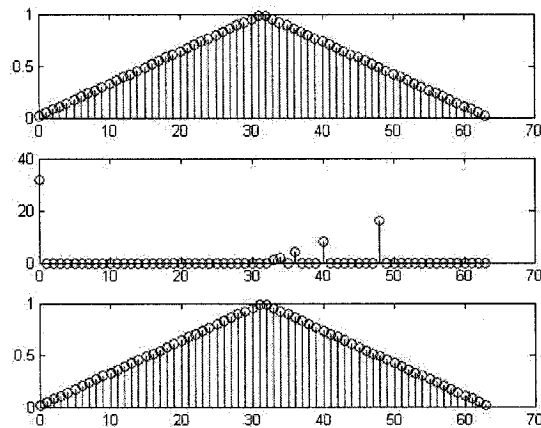


Figure 5.7: Transformation d'un signal triangulaire par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit.

Signal de rampe : $x[n] = n$

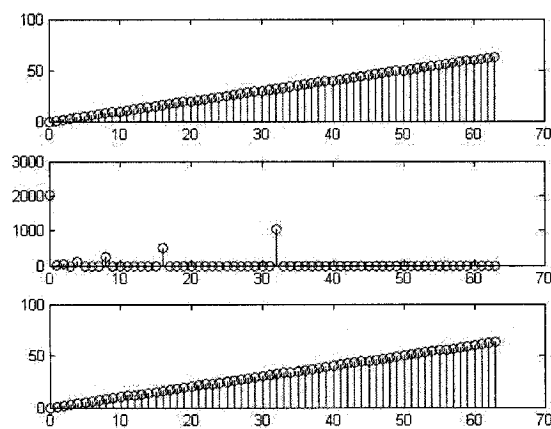


Figure 5.8: Transformation d'un signal de rampe par la TDWH.

Le premier graphe : le signal d'entrée. Le deuxième : le signal dans le domaine de Fourier.
Le troisième le signal reconstruit

Signal bidimensionnel : $x[n] = image$



Figure 5. 9: Résultat de la transformation d'une image par la TDWH

6. Transformée Discrète Fractionnelle de Fourier (TDFrF)

La Transformée Fractionnelle de Fourier a été introduite dans la littérature mathématique dans les années 80 [11, 12]. Les chercheurs ont commencé à l'utiliser comme une technique de traitement des signaux dans les années 90, par Almeida et al [13, 14].

6.1.Définition :

Si on considère un signal $x(t)$ représenté le long de l'axe du temps et sa transformée de Fourier $X(w)$ représentée le long de l'axe de fréquence, nous pouvons maintenant regarder l'opérateur de Transformée de Fourier \mathbb{F} comme une rotation dans le sens contraire de l'aiguille d'une montre par un angle de $\pi/2$. Cet opérateur se conforme avec les résultats de l'application répétées de la transformée de Fourier, comme le montrent les équations suivantes :

$$\begin{aligned}\mathbb{F}x(t) &= X(w) \\ \mathbb{F}\mathbb{F}x(t) &= x(-t) \\ \mathbb{F}\mathbb{F}\mathbb{F}x(t) &= X(-w) \\ \mathbb{F}\mathbb{F}\mathbb{F}\mathbb{F}x(t) &= x(t)\end{aligned}\tag{6.1}$$

Le but de la Transformée Fractionnelle de Fourier est de représenter le signal dans un domaine entre le temps et la fréquence en effectuant une rotation du signal par un angle quelconque α figure (6.1).

La fonction de base de cette transformée est donnée par l'équation suivante [13]

$$K_{\alpha}(t,u) = \begin{cases} (1) \rightarrow \sqrt{\frac{1-j\cot\alpha}{2\pi}} e^{j\frac{t^2+u^2}{2}\cot\alpha - jut\csc\alpha} \\ (2) \rightarrow \delta(t-u) \\ (3) \rightarrow \delta(t+u) \end{cases}\tag{6.2}$$

(1) si α n'est pas un multiple de π

(2) si α est un multiple de 2π

(3) si $\alpha + \pi$ est un multiple de 2π

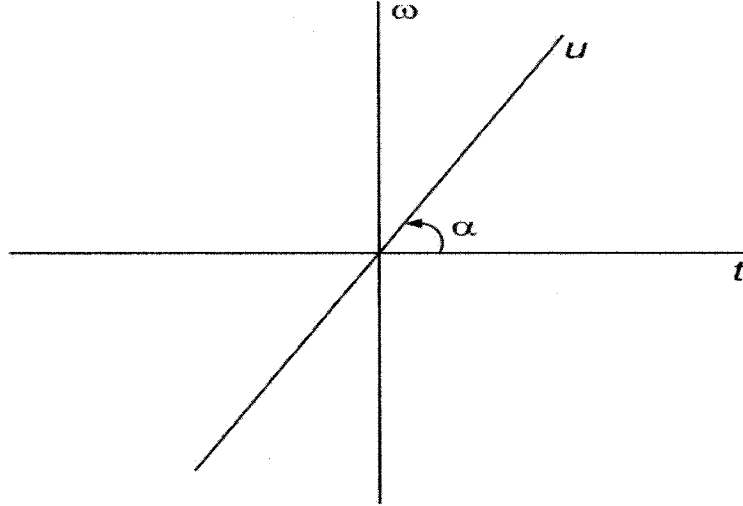


Figure 6. 1: Domaine de la transformation de la transformée fractionnelle d'ordre α

Cette fonction de base est obtenue de celle de la Transformée Linéaire Canonical (TLC). La TLC est une forme généralisée de la Transformée Fractionnelle de Fourier (TFrF) [60].

La Transformée Linéaire Canonical (TLC) est donnée par l'équation suivante :

$$TLC\{f(t)\} = \int_{-\infty}^{+\infty} C_M(u,t) f(t) dt \quad (6.3)$$

où

$$C_M(u,t) = \sqrt{\beta'} \exp(-j\pi/4) \exp(j\pi(\alpha' u^2 - 2\beta' ut + \gamma' t^2)) \quad (6.4)$$

Les paramètres $(\alpha', \beta', \gamma')$ sont indépendants de u et t . Ces paramètres sont groupés dans une matrice M . La matrice M sera le paramètre de la Transformée Linéaire Canonical (TLC). La relation entre les composantes de la matrice M et les paramètres $(\alpha', \beta', \gamma')$ est donnée par l'équation suivante :

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{\gamma'}{\beta'} & \frac{1}{\beta'} \\ -\beta' + \frac{\alpha'\gamma'}{\beta'} & \frac{\alpha'}{\beta'} \end{bmatrix} \quad (6.5)$$

En calculant les paramètres $(\alpha', \beta', \gamma')$ en fonction des composantes de la matrice M , on trouve que

$$\alpha' = \frac{D}{B} \quad (6.6)$$

$$\beta' = \frac{1}{B} \quad (6.7)$$

$$\gamma' = \frac{A}{B} \quad (6.8)$$

La rotation de 2-dimension par un angle α quelconque dans le sens contraire à l'aiguille d'une montre est donnée par cette matrice :

$$R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (6.9)$$

Soit $M=R$, on obtient

$$A = \cos(\alpha); \quad B = \sin(\alpha); \quad C = -\sin(\alpha); \quad D = \cos(\alpha). \quad (6.10)$$

et les paramètres $(\alpha', \beta', \gamma')$ seront égales à :

$$\alpha' = \cot(\alpha); \quad \beta' = \csc(\alpha); \quad \gamma' = \cot(\alpha). \quad (6.11)$$

En remplaçant α', β' et γ' par leurs valeurs dans l'équation (6.4), la fonction de base de la Transformée Linéaire Canonique (TLC) sera égale à celle de la Transformée Fractionnelle de Fourier.

La Transformée Fractionnelle de Fourier d'un signal $x(t)$ est :

$$TFrF\{x(t)\} = X_\alpha(u) = \int_{-\infty}^{+\infty} x(t) K_\alpha(t, u) dt \quad (6.12)$$

La transformée inverse d'une transformation fractionnelle avec un angle α est une transformation fractionnelle avec un angle $-\alpha$

$$TFrFI\{X_\alpha(u)\} = x(t) = \int_{-\infty}^{+\infty} X_\alpha(u) K_{-\alpha}(u, t) du \quad (6.13)$$

6.1.1 Transformée Discrète Fractionnelle de Fourier (TDFrF) :

Une fois que la Transformée Fractionnelle continue a été définie, plusieurs chercheurs ont essayé de dériver la forme numérique ou discrète de cette transformée. Vu que le domaine de cette transformée est mixte, les chercheurs ont suggéré plusieurs formes différentes.

Nous tenterons de décrire brièvement ces formes dans le but de les comparer et d'en tirer la plus convenable pour le traitement des signaux.

Dans notre comparaison, la forme la plus appropriée sera celle qui répond aux critères suivants : 1) la matrice devrait être réversible pour assurer l'existence de la transformée inverse. 2) La matrice devrait être orthogonale pour être une de la famille des transformée orthogonales. 3) Les résultats de son application devraient être conformes avec celles de la Transformée Fractionnelle continue.

La forme discrète la plus simple de cette transformée est obtenue par l'échantillonnage et la computation directe de la forme continue [28]. De cette façon, la forme résultante perdrait plusieurs propriétés importantes telle que la réversibilité et l'addition ce qui limite son application dans le traitement des signaux. Ozaktas et al,[29], ont proposés une façon nouvelle pour l'échantillonnage de la forme continue dans le but d'améliorer sa performance et la rendre plus utile. Bien que le résultat de cette forme soit plus conforme avec la forme continue, sa matrice n'est ni orthogonale ni additive. Ce qui ajoute encore plus de contraintes sur le signal d'entrée. Santhanam et al, [30] ont proposé un algorithme pour calculer la Transformée Discrète Fractionnelle de Fourier par une composition linéaire. Cette composition linéaire contient un mélange angulaire du signal et sa Transformée Discrète de Fourier avec un renversement circulaire du signal et de sa Transformée Discrète de Fourier. Par cet algorithme la matrice de la transformée est orthogonale, réversible, additive. Cependant, les résultats émanant de cette forme ne ressemblent pas à ceux de la forme continue. D'autres chercheurs [31, 32, 33] ont proposé de trouver les vecteurs et les valeurs propres de la matrice de la Transformée Discrète de Fourier et ensuite calculer sa puissance fractionnelle. Cela résulte en une matrice orthogonale, réversible et additive aussi performante que la forme continue de la transformée. Cependant, il n'existe aucun algorithme rapide qui la calcule efficacement ce qui limite son utilisation dans le domaine de traitement des signaux. Plusieurs autre formes ont été introduites récemment [34, 45] dont la transformée satisfait les propriétés de l'addition, de la réversibilité et ressemble à la

forme continue, mais ces formes ne sont pas définies pour toutes les valeurs de α , l'angle de rotation.

Pei et al [28] ont dérivés une forme de la Transformée Discrète Fractionnelle de Fourier en débutant par la définition de la transformée Affine de Fourier qui est la forme généralisée de notre transformée fractionnelle.

On considère cette forme comme la forme la plus convenable pour le traitement des signaux parce qu'elle satisfait à la majorité des propriétés outre l'addition. Elle permet aussi de fournir des résultats conformes avec ceux de la forme continue et plusieurs algorithmes rapides ont été mis en place pour son implantation. Pei a surmonté l'inconvénient de l'addition en fournissant une relation qui convertit les résultats de la transformée d'un angle α_1 en un angle α_2 sans passer par la transformée inverse.

La forme générale de cette transformée discrète est définie par l'équation suivante :

$$X_{(p,q)}[m] = \sqrt{\frac{1}{2M+1}} \exp(jm^2q/2) \sum_{n=-N}^N x[n] \exp(-j2\pi nm/2M+1) \exp(jn^2p/2) \quad (6.14)$$

où

$$M \geq N \quad (6.15)$$

et

$$\begin{aligned} p &= \cot(\alpha) \Delta u^2 \\ q &= \cot(\alpha) \Delta t^2 \end{aligned} \quad (6.16)$$

et sa transformée inverse :

$$x_{(p,q)}[n] = \sqrt{\frac{1}{2M+1}} \exp(-jn^2p/2) \sum_{m=-N}^N X_{(p,q)}[m] \exp(j2\pi nm/2M+1) \exp(-jm^2q/2) \quad (6.17)$$

Les paramètres p et q dans la définition contrôlent la multiplication de la fonction de "Chirp" dans le domaine temporel et fréquentiel respectivement. Si on choisit $p=q$ et $M=N$, la matrice de la Transformée Discrète Fractionnelle de Fourier sera symétrique et orthogonale. La transformée inverse sera la conjuguée de la matrice de la transformée à sens directe. À ce stade, on peut donc définir la transformée

fractionnelle et son inverse par les équations suivantes après avoir effectué les changements sur les paramètres p , q , N et M :

$$X_{(p)}[m] = \sqrt{\frac{1}{2N+1}} \exp(jm^2 p / 2) \sum_{n=-N}^N x[n] \exp(-j2\pi nm / 2N+1) \exp(jn^2 p / 2) \quad (6.18)$$

$$x_{(p)}[n] = \sqrt{\frac{1}{2N+1}} \exp(-jn^2 p / 2) \sum_{m=-N}^N X_{(p)}[m] \exp(j2\pi nm / 2N+1) \exp(-jm^2 p / 2) \quad (6.19)$$

6.2. Transformée Discrète Bidimensionnelle Fractionnelle de Fourier

La transformée bidimensionnelle directe est décrite par l'équation suivante :

$$X_{(p)}[v, u] = \frac{1}{2N+1} \exp(j(u^2 + v^2) p / 2) \sum_{m=-N}^N \sum_{n=-N}^N x[m, n] \exp(-j2\pi(nu + mv) / 2N+1) \exp(j(n^2 + m^2) p / 2) \quad (6.20)$$

et son inverse par l'équation suivante :

$$x_{(p)}[m, n] = \frac{1}{2N+1} \exp(-j(n^2 + m^2) p / 2) \sum_{v=-N}^N \sum_{u=-N}^N X_{(p)}[v, u] \exp(j2\pi(nu + mv) / 2N+1) \exp(-j(u^2 + v^2) p / 2) \quad (6.21)$$

Comme la fonction de base de notre transformée est séparable, alors l'algorithme de ligne colonne de la transformation bidimensionnelle peut être appliqué pour définir la transformée bidimensionnelle discrète fractionnelle de Fourier.

En réécrivant l'équation (6.20) sous la forme suivante :

$$X_{(p)}[m] = F_{(p)} x[n] \quad (6.22)$$

où

$$F_{(p)} = \sqrt{\frac{1}{2N+1}} \exp(jm^2 p / 2) \exp(-j2\pi nm / 2N+1) \exp(jn^2 p / 2) \quad (6.23)$$

La transformée bidimensionnelle après l'application de l'algorithme ligne colonne sera :

$$X_{(p)}[v, u] = F_{(p)} x[m, n] F_{(p)}^{-1} \quad (6.24)$$

où $F_{(p)}$ et $F_{(p)}^{-1}$ sont les matrices de la transformée fractionnelle d'ordre (p) et la transformée fractionnelle inverse d'ordre (p) respectivement.

et son inverse est :

$$x_{(p)}[m, n] = F_{(p)}^{-1} X_{(p)}[v, u] F_{(p)} \quad (6.25)$$

6.3. Propriétés de la Transformée Discrète Fractionnelle de Fourier :

Symétrie :

Pour que la matrice de la transformée soit symétrique il faut que :

$$F_{(p)}[m, n] = F_{(p)}[n, m] \quad (6.26)$$

En examinant l'équation de la matrice, on réalise que si on change m par n et n par m dans l'équation (6.23), le résultat de ce changement n'affecte pas l'équation et elle reste toujours égale à l'équation de départ.

Conjugaison :

En comparant les deux équations (6.20) et (6.21) de la matrice de la transformée de sens direct et de sens inverse respectivement, on réalise qu'on peut déduire l'une de l'autre en prenant les conjugués de leurs fonctions exponentielles.

$$F_{(p)}^* = F_{(p)}^{-1} \quad (6.27)$$

Renversement temporel :

Dans l'équation (6.20) est définie la transformée fractionnelle dans le sens direct si on change n par -n, on aperçoit que le seul changement est fait sur l'exponentiel commun entre n et m et les autres exponentiels restent intacts. Par similarité aux propriétés de renversement temporel de la Transformée Discrète de Fourier, on peut dire :

Si

$$X_{(p)}[m] = TDFrF \{x[n]\} \quad (6.28)$$

Alors

$$TDFrF \{x[-n]\} = X_{(p)}[-m] \quad (6.29)$$

Conservation de l'énergie :

Pour que cette transformée conserve l'énergie, elle doit obéir au théorème de Parseval :

$$\sum_{m=-N}^N |X_{(p)}[m]|^2 = \sum_{n=-N}^N |x[n]|^2 \quad (6.30)$$

Preuve :

$$\begin{aligned} \sum_{m=-N}^N |X_{(p)}[m]|^2 &= \sum_{m=-N}^N X_{(p)}[m] X_{(p)}^*[m] \\ &= \sum_{m=-N}^N \sum_{n=-N}^N \sum_{l=-N}^N F_{(p)}[m] F_{(p)}^*[m] x[n] x^*[l] \\ &= \sum_{n=-N}^N \sum_{l=-N}^N \delta(n-l) x[n] x^*[l] \\ &= \sum_{n=-N}^N |x[n]|^2 \end{aligned} \quad (6.31)$$

Pour une liste complète de toutes les propriétés vous pouvez vous référer à l'article de Pei [28].

6.4. Orthogonalité de la Transformée Discrète Fractionnelle de Fourier

Pour que cette transformée soit orthogonale, on doit démontrer que

$$\langle F_{2N+1}^{nm_1}, F_{2N+1}^{nm_2} \rangle = \begin{cases} c, & m_1 = m_2 \\ 0, & m_1 \neq m_2 \end{cases} \quad (6.32)$$

où c est une constante et

$$F_{2N+1}^{nm_1} = \exp(jm_1^2 p / 2) \exp(-j2\pi nm_1 / 2N + 1) \exp(jn^2 p / 2) \quad (6.33)$$

$$F_{2N+1}^{nm_2} = \exp(jm_2^2 p / 2) \exp(-j2\pi nm_2 / 2N + 1) \exp(jn^2 p / 2) \quad (6.34)$$

et la fonction $\langle x, y \rangle$ est définie par l'équation (2.58). Alors

$$\begin{aligned} \langle F_{2N+1}^{nm_1}, F_{2N+1}^{nm_2} \rangle &= \sum_{n=0}^{2N} \exp(jm_1^2 p / 2) \exp(-j2\pi nm_1 / 2N + 1) \exp(jn^2 p / 2) \\ &\quad \exp(-jm_2^2 p / 2) \exp(j2\pi nm_2 / 2N + 1) \exp(-jn^2 p / 2) \end{aligned} \quad (6.35)$$

$$\langle F_{2N+1}^{nm_1}, F_{2N+1}^{nm_2} \rangle = \sum_{n=0}^{2N} \exp(j(m_1^2 - m_2^2) p / 2) \exp(-j2\pi n(m_1 - m_2) / 2N + 1) \quad (6.36)$$

Il y a deux cas à étudier :

$$1) m_1 = m_2$$

$$\langle F_{2N+1}^{nm_1}, F_{2N+1}^{nm_2} \rangle = \sum_{n=0}^{2N} \exp(-j2\pi n(0)/2N+1) \exp(j(0)p/2) = 2N+1 \quad (6.37)$$

2) $m_1 \neq m_2$

$$\begin{aligned} \langle F_{2N+1}^{nm_1}, F_{2N+1}^{nm_2} \rangle &= \sum_{n=0}^{2N} \exp(jm_3 m_4 p/2) \exp(-j2\pi n(m_4)/2N+1) \\ &= \exp(jm_3 m_4 p/2) \sum_{n=0}^{2N} \exp(-j2\pi n(m_4)/2N+1) = 0 \end{aligned} \quad (6.38)$$

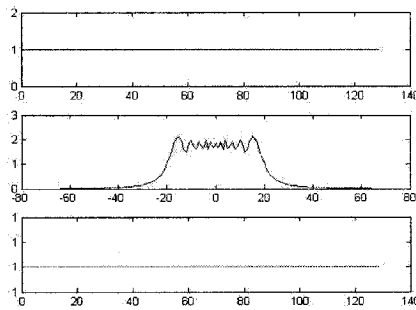
où

$$\begin{aligned} m_3 &= m_1 + m_2 \\ m_4 &= m_1 - m_2 \end{aligned} \quad (6.39)$$

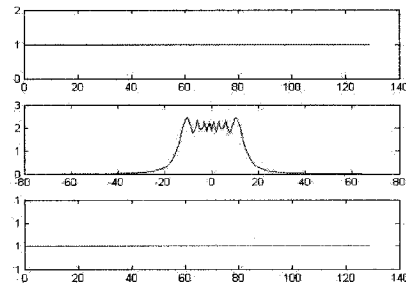
6.5. Application de la TDFrF à une classe de signaux :

La Transformée Discrète Fractionnelle de Fourier (TDFrF) et son inverse sont programmés pour calculer et afficher les résultats de la classe des signaux du tableau (2.2). Cette étude est faite pour plusieurs ordres.

Signal d'échelon : $x[n] = u[n]$



(a)



(b)

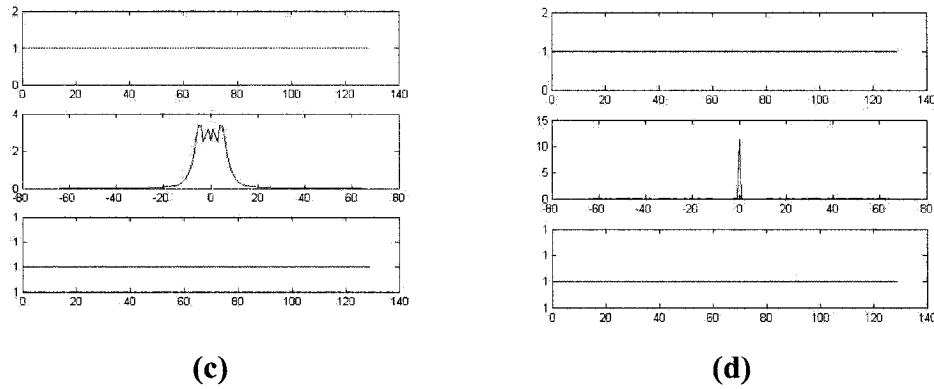


Figure 6. 2: Transformation d'un signal d'échelon par la TDFrF.

(a) $\alpha = 0.4\pi/2$.(b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

Signal rectangulaire : $x[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{autrement} \end{cases}$

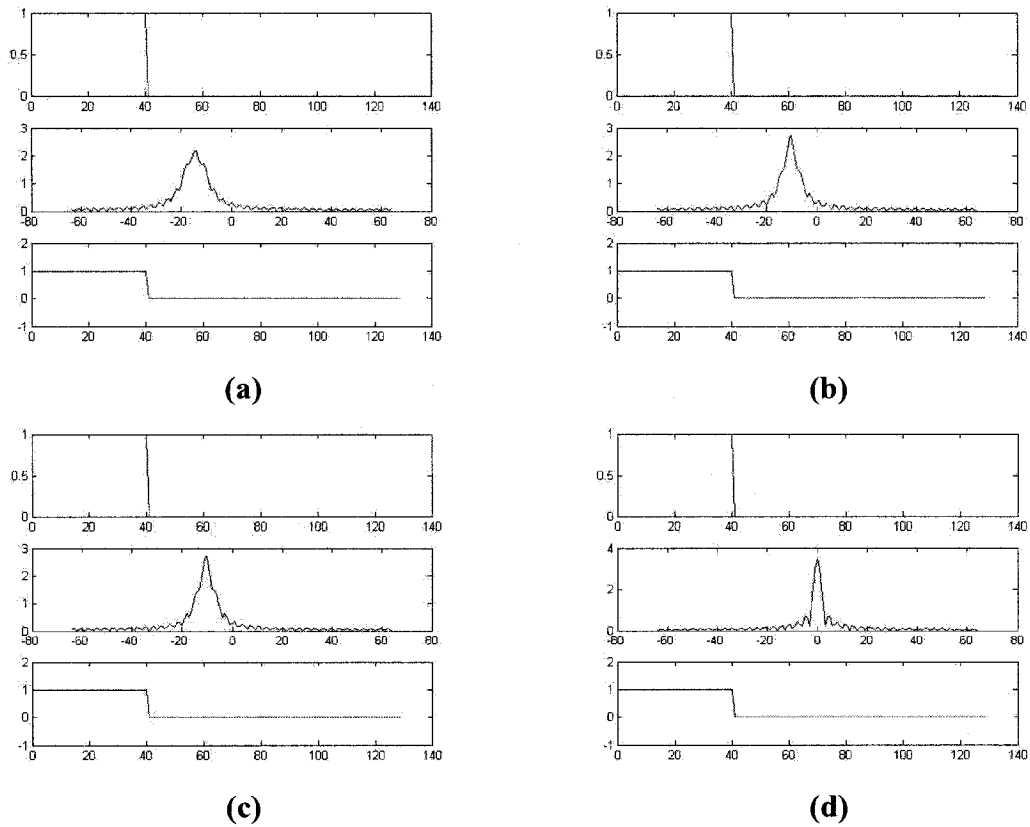
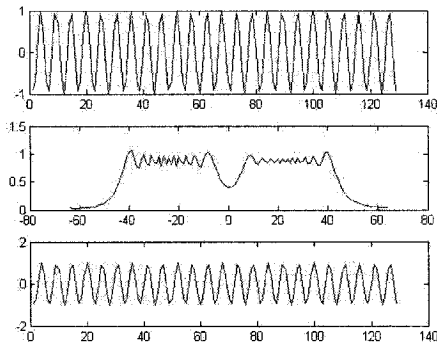


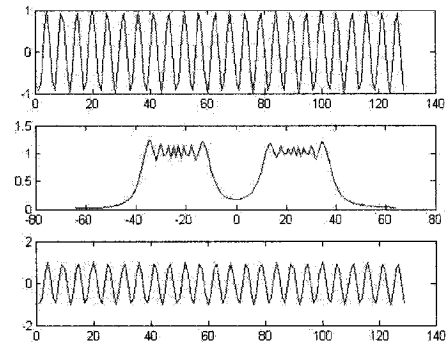
Figure 6. 3: Transformation d'un signal rectangulaire par la TDFrF.

(a) $\alpha = 0.4\pi/2$.(b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

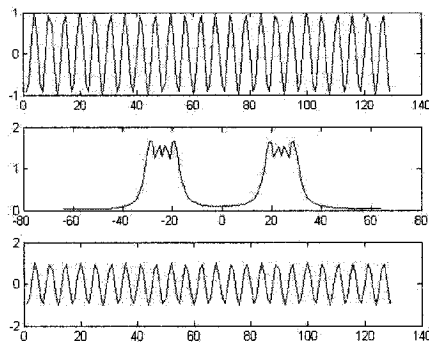
Signal sinusoidal : $x[n] = \sin(2\pi fn)$



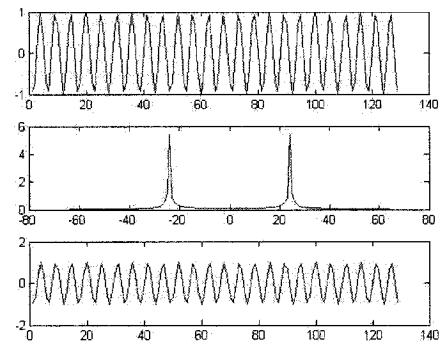
(a)



(b)



(c)

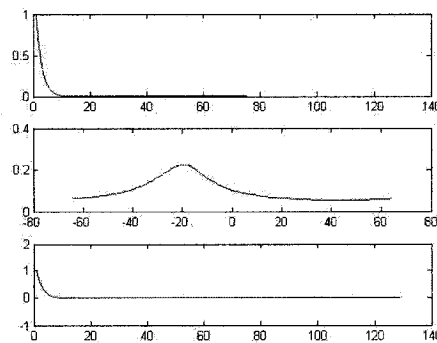


(d)

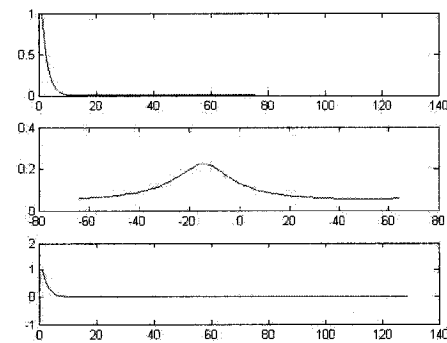
Figure 6. 4: Transformation d'un signal sinusoidal par la TDFrF.

(a) $\alpha = 0.4\pi/2$. (b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

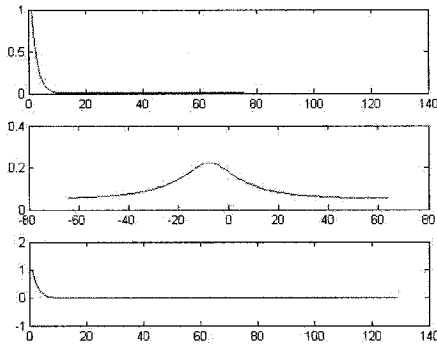
Signal exponentiel : $x[n] = \exp(-\alpha n)$



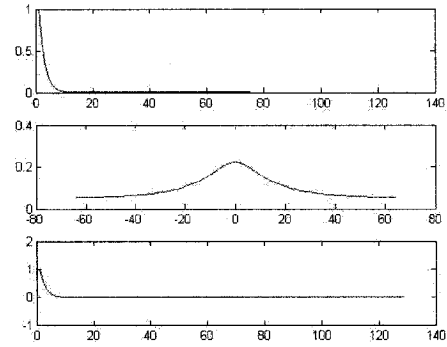
(a)



(b)



(c)

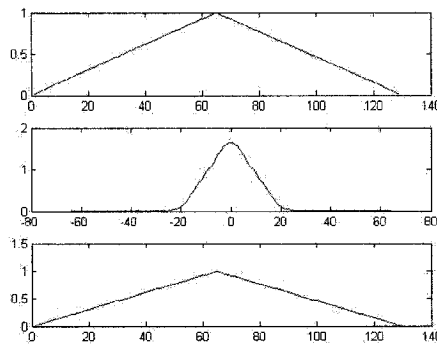


(d)

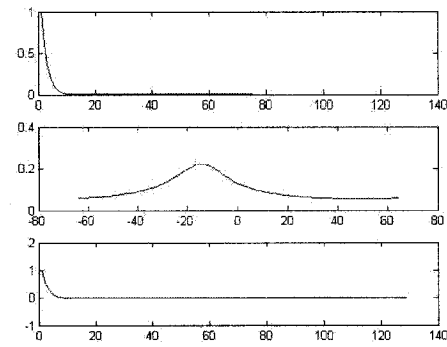
Figure 6. 5: Transformation d'un signal exponentiel par la TDFrF.

(a) $\alpha = 0.4\pi/2$. (b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

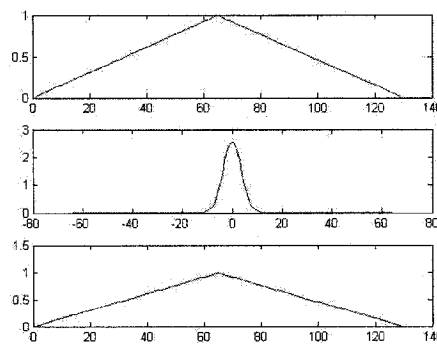
Signal triangulaire : $x[n] = \Lambda_N$



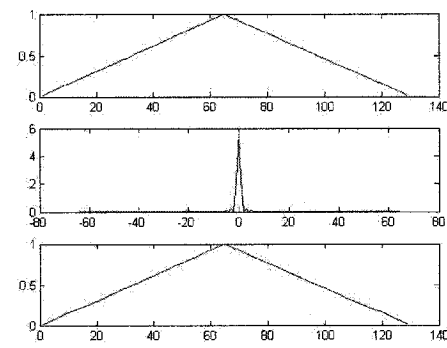
(a)



(b)



(c)

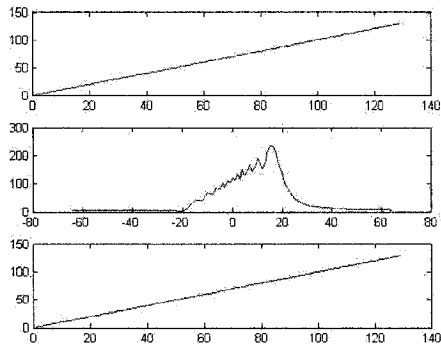


(d)

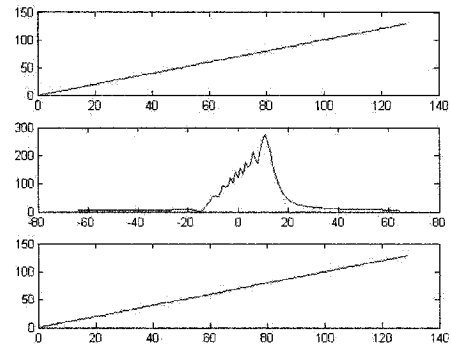
Figure 6. 6: Transformation d'un signal triangulaire par la TDFrF.

(a) $\alpha = 0.4\pi/2$. (b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

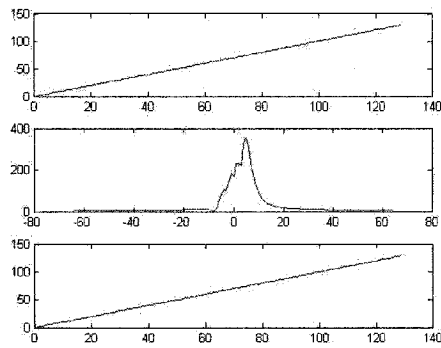
Signal de rampe : $x[n] = n$



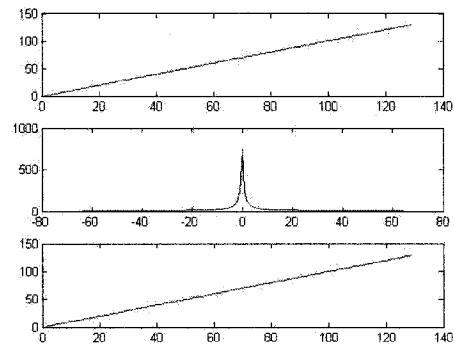
(a)



(b)



(c)

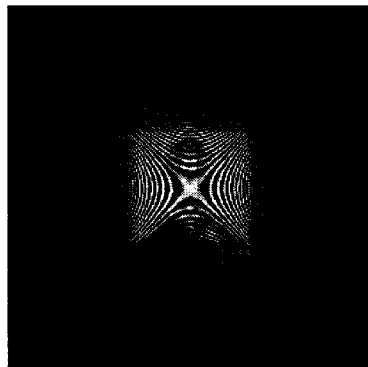


(d)

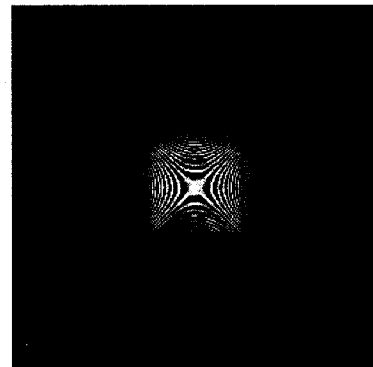
Figure 6. 7: Transformation d'un signal de rampe par la TDFrF.

(a) $\alpha = 0.4\pi/2$. (b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

Signal bidimensionnel : $x[n] = \text{image}$



(a)



(b)



Figure 6. 8: Résultat de la transformation d'une image par la TDFrF.

(a) $\alpha = 0.4\pi/2$. (b) $\alpha = 0.6\pi/2$. (c) $\alpha = 0.8\pi/2$. (d) $\alpha = \pi/2$

7. Application à la Reconnaissance de la parole

La parole est produite par le passage d'air à travers plusieurs obstructions et cheminements, comme le larynx, la gorge, la bouche, la langue, les lèvres etc., dans le système vocal de l'être humain. Elle est émise sous la forme d'une série d'ondes. Pour convertir ces ondes en des mots écrits, une série d'opérations doit être appliquée. Ces opérations contiennent l'acquisition et la représentation des ondes sous forme de notations appropriées, la production de vecteurs qui représentent les tranches temporelles de l'entrée convertie, le groupement et la purification d'entrée. Cela est suivi par un assortiment des résultats contre une bibliothèque de vecteurs de phonèmes connus, le choix de la série de lettre la plus susceptible et finalement l'organisation des mots écrits d'une façon significative.

7.1. Système de la reconnaissance de la parole :

Dans chaque stade d'opération, il y a plusieurs techniques qui peuvent être utilisées pour atteindre notre but. Pour la représentation des ondes de la parole, des Transformées reliées à Fourier ont été utilisées par plusieurs chercheurs.

La figure ci-dessous montre un diagramme fonctionnel général d'un système de reconnaissance de la parole. Par la suite on décrira brièvement chaque bloc fonctionnel et ensuite on élaborera le rôle de la Transformée de Fourier et ses transformées reliées dans la représentation du signal de la parole.

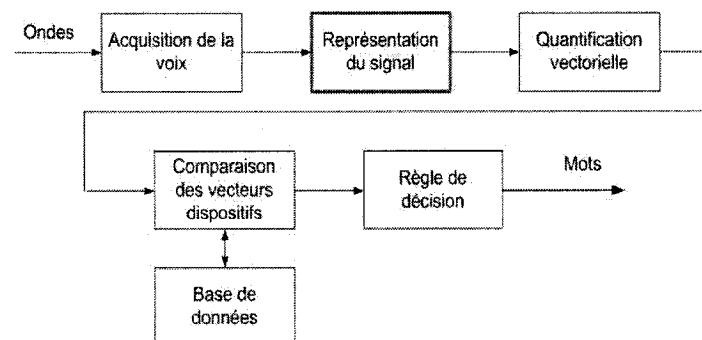


Figure 7. 1: Diagramme Fonctionnel général d'un système de reconnaissance de la parole.

Acquisition de la voix :

La parole est saisie à l'aide du microphone qui transforme les ondes propagées, en des signaux. Ces derniers peuvent être enregistrés pour un traitement direct ou ultérieur. La figure (7.2) représente un signal vocal saisi par un microphone.

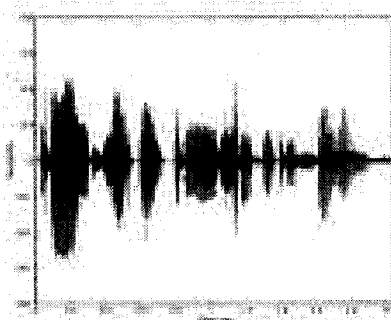


Figure 7. 2 : Signal vocal d'une personne.

Le signal de la parole passe dans un filtre qui élimine et réduit le bruit pour assurer que le signal contient seulement la parole. On trouve plusieurs techniques et systèmes de filtrage qui sont utilisés dans les systèmes de la reconnaissance de la parole. Parmi ces techniques on cite le système 'Voice Recognition Enhancer' [61]. Ce système fournit une réduction de bruit de 6 à 18 dB sans affecter ou détruire les éléments de la parole dans le signal.

Représentation du signal :

Le but de ce bloc est de représenter le signal de la parole d'une façon on peut retirer les composantes fréquentielles du signal et le temps associé à chacune. Parmi les techniques utilisées pour atteindre ce but on cite le Codage Prédicatif Linéaire (CPL) et l'Analyse Cepstrale. Le Codage Prédicatif Linéaire se base sur une supposition que la parole est produite par un 'buzzer' au bout d'un tube. Le CPL analyse le signal de la parole en estimant les 'formants' qui sont produites par le tube et ils sont caractérisés par leur résonance. Après l'estimation de ces formants le CPL réduit leur influence sur le signal de la parole. Le CPL ensuite estime les 'Buzz' dans le signal de la parole qui sont caractérisés par leur intensité et leur fréquence. Comme le signal de la parole varie avec le temps le CPL est appliqué sur des petits intervalles de temps.

L'analyse Cepstrale consiste à calculer la Transformée de Fourier de petites fenêtres de la parole, décorrélér le spectre obtenu à l'aide de la Transformée Discrète en Cosinus et ensuite choisir la première composante significative.

Quantification Vectorielle :

Dans ce bloc, les vecteurs obtenus du bloc précédent sont groupés ensemble pour former des 'Clusters'. Ce groupage de données est connu sous le nom 'Partitionnement de données'. L'idée de ce groupage est de partitionner les données en sous-ensembles dont les vecteurs possèdent des traits communs. Les vecteurs qui appartiennent à un sous-ensemble, sont représentés par le vecteur du centre de leur sous-ensemble. Parmi les techniques utilisées pour la partitionnement de données on cite le 'K-means' et le 'Fuzzy c-means'. Ces vecteurs obtenus sont quantifiés pour réduire leur taille.

Comparaison des vecteurs dispositifs et Base de données :

Après la Quantification Vectorielle, les vecteurs dispositifs obtenus seront assortis à une base de données qui contient des vecteurs représentants toutes les phonèmes de la langue pour différents orateurs et par différentes prononciations.

Règle de décision :

Les vecteurs les plus probables obtenus du bloc précédent passe au dernier bloc où la langue est modélisée. La modélisation de la langue a le but de choisir les mots d'une façon significative et de réduire le taux d'erreur.

7.2.Transformée de Fourier dans la reconnaissance de la parole :

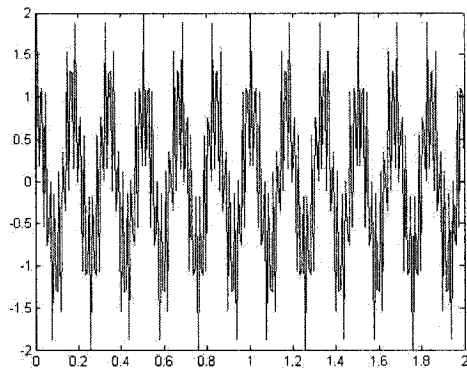
Dans la figure (7.2), on remarque qu'on ne peut pas déduire la parole dite en examinant le signal. Puisque les composantes fréquentielles d'un signal sont les informations les plus importantes et nécessaires pour l'identification de la voix, alors une transformation qui produit ces composantes fréquentielles est nécessaire. La technique la plus commune est la Transformée de Fourier. La Transformée de Fourier n'est qu'une représentation du signal sous forme de fréquences et d'amplitudes codifiés dans le signal.

La Transformée de Fourier nous fournit une relation entre le temps, la fréquence et l'amplitude qui est une combinaison parfaite pour la reconnaissance. Cependant, la

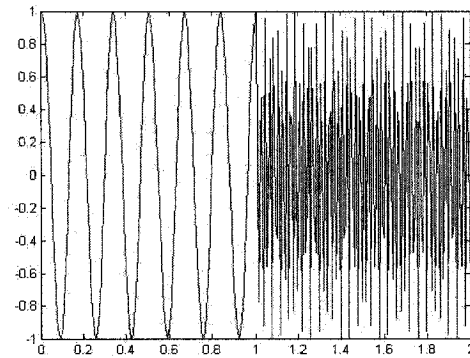
Transformée de Fourier manque un facteur très important pour la reconnaissance de la parole : le temps associé à chacune de ces composantes fréquentielles. En effet, la Transformée de Fourier ne donne aucune indication à quel temps ces composantes fréquentielles se produisent. Elle considère que chaque fréquence existe pendant tout l'intervalle de temps. Pour montrer le problème d'une façon plus claire, considérons deux signaux différents x_1 et x_2

$$x_1 = \cos(w_1 t) + \cos(w_2 t) \rightarrow 0 \leq t \leq 2s \quad (7.1)$$

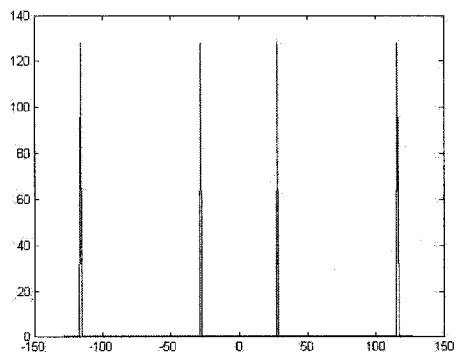
$$x_2 = \begin{cases} \cos(w_1 t) \rightarrow 0 \leq t \leq 1s \\ \cos(w_2 t) \rightarrow 1 \leq t \leq 2s \end{cases} \quad (7.2)$$



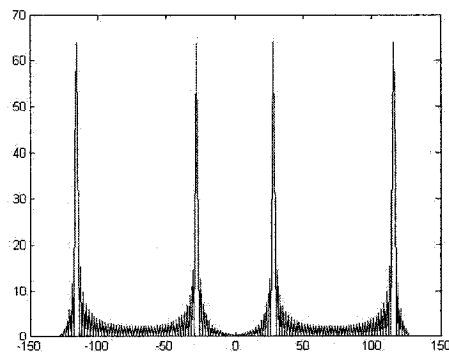
(a)



(b)



(c)



(d)

Figure 7.3 : Performance de la transformée de Fourier sur des signaux stationnaire et non stationnaire.
 (a) Graphique de x_1 dans le temps, (b) graphique de x_2 dans le temps, (c) transformée de Fourier de x_1 et (d) transformée de Fourier de x_2

La Transformée de Fourier de x_1 et x_2 donne des résultats identiques même si les deux signaux d'entrée sont différents, figure (7.2). Ceci fait en sorte que la Transformée de Fourier ne peut être utilisée qu'avec des signaux dont les fréquences existent pendant tout l'intervalle de temps, ou en d'autres termes avec les signaux stationnaires. Cependant, les signaux de la parole sont rarement stationnaires et les fréquences existent dans de petits intervalles de temps. La transformée de Fourier traditionnelle, telle que présentée jusqu'à ce point n'est donc pas appropriée pour la reconnaissance de la parole.

Les chercheurs ont surmonté ce problème en introduisant de nouvelles transformées qui sont reliées à la transformée de Fourier comme la Transformée de Fourier à Court Temps [36] et la Transformée Fractionnelle de Fourier [37].

La première solution décompose le signal en petits intervalles de temps de N points à l'aide de fenêtres. Pendant chaque intervalle on considère que le signal est stationnaire ce qui nous permet d'appliquer la transformée de Fourier :

$$x[n] = s[n]w[n] \rightarrow n = 0, 1 \dots N \quad (7.3)$$

où $x[n]$ est le signal dans un intervalle de N points, $s[n]$ est notre signal de la parole, $w[n]$ est la fenêtre utilisée, limitée à N points.

La TFCT (Transformée de Fourier à Court Temps) nous donne la réponse fréquentielle $X[k]$ de $x[n]$. $X[k]$ est la convolution entre la transformée de Fourier de $s[n]$ et celle de $w[n]$.

Ceci nous donnerait les fréquences d'un signal et leurs endroits dans le temps. Cette analyse est connue sous le nom « analyse temps-fréquence ».

La transformée de Fourier à court temps a résolu le problème de l'association temps-fréquence mais a introduit un nouveau dilemme sur comment choisir la fenêtre la plus convenable pour la reconnaissance de la parole.

Plusieurs chercheurs ont étudié et comparé les fenêtres traditionnelles comme la fenêtre rectangulaire, la fenêtre triangulaire, la fenêtre de Blackman, la fenêtre de Hanning, la fenêtre de Hamming etc., pour déduire quelle fenêtre est la meilleure

pour la représentation des signaux pour la reconnaissance de la parole. Les critères sur lesquels les chercheurs se sont basés, sont des critères de filtrage en général : l'amplitude du signal après le passage par la fenêtre doit être le plus proche de celle du signal avant le passage par la fenêtre. Ceci s'ajoute à la résolution spectrale et l'étalement spectral des fenêtres.

D'après [38], la fenêtre de Hamming est la plus convenable. Ces conclusions sont basées sur la forme de la transformée de Fourier de cette fenêtre; figure (7.3), où la largeur du lobe principale est grand et les lobes sous principaux sont moins grands.

Rozman et al [39], est allé plus loin en posant la question suivante

"Est ce que les critères choisis sont les meilleurs pour la reconnaissance de la parole ?"

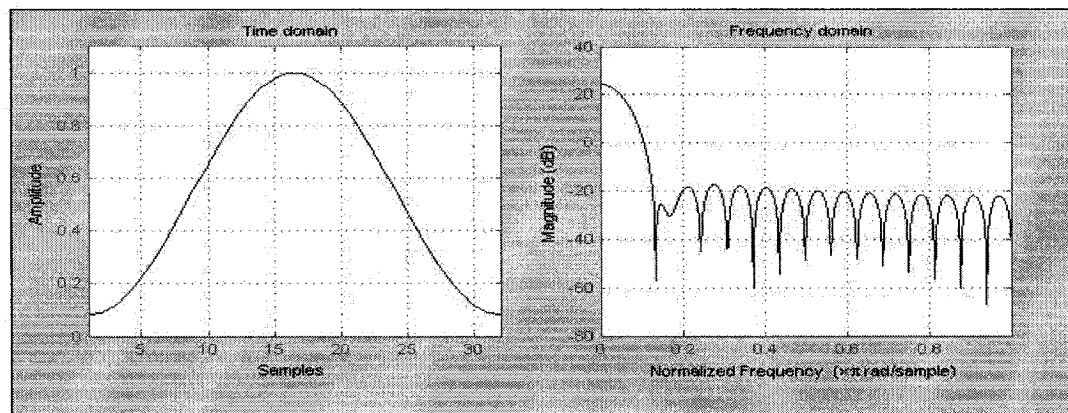


Figure 7. 4: Fenêtre de Hamming et sa transformée de Fourier.

Le graphique à gauche est la fenêtre de Hamming dans le temps cela à droite est la transformée de Fourier de la fenêtre de Hamming.

Pour répondre à cette question, il a pris comme référence le système auditoire de l'être humain pour développer de nouveaux critères pour les fenêtres. En se basant sur ces nouveaux critères, Rozman a proposé un design pour les filtres RIF. Ce design donne des filtres avec une erreur minimale selon le critère de Chebysheve. Ces nouveaux filtres sont maintenant utilisés comme des fenêtres dans la Transformée de Fourier à Court Temps.

La méthode de la deuxième transformée ; la Transformée Fractionnelle de Fourier qui a été décrite dans le chapitre (6) représente le signal de la parole dans le plan temps fréquence après une rotation par un angle α quelconque plus petit que $\pi/2$.

Par cette rotation du signal de la parole, les composantes fréquentielles du signal et leurs endroits dans le temps seront explicites à l'analyseur.

Problème de ces solutions :

Pour bien représenter un signal dans le plan temps fréquence, il faut que la résolution dans le temps soit assez grande (pour qu'elle puisse contenir plus qu'une période) pour les basses fréquences et petite pour les hautes fréquences. Les deux transformées présentées ci-dessus ne peuvent être utilisées qu'avec des résolutions fixes. Les fenêtres doivent être de même largeur et la rotation des axes doit être faite sur le long de l'axe et pour toutes les valeurs.

Pour cette raison, la Transformée en Ondelettes [40] a été suggérée comme une technique de représentation parce qu'elle transforme le signal au plan temps fréquence avec une résolution variable.

8. Application à la compression des images

8.1.Introduction

Avec l'évolution rapide des ordinateurs et l'utilisation excessive des images, les chercheurs ont travaillé sur des algorithmes qui les compressent. Ces algorithmes ont but de réduire l'espace de leur enregistrement dans la mémoire ou de faciliter leur transmission. Ces algorithmes sont divisés en deux familles : les algorithmes de compression sans perte d'information et les algorithmes avec pertes d'informations.

Algorithmes sans pertes d'informations :

Les algorithmes sans pertes d'informations doivent obéir au théorème de Shannon [41], qui stipule qu'il y a des limites sur la compression qui doivent être respectées pour qu'on ne perde pas d'informations.

Parmi ces algorithmes, on retrouve les algorithmes de Lempel-Ziv [42, 43, 44] qui se fondent sur la représentation des séquences de bit du signal d'entrée. Les séquences fréquentes sont représentées par un petit mot de code et les séquences moins probables par des mots de code plus grands.

Deux autres techniques probabilistes ou codage d'entropies telles que l'algorithme Arithmétique [45] et le codage de Huffman [46] existent. La première commence à déterminer un modèle prédictif de données ou une prédiction des patrons (patterns) qui se trouvent dans le message qu'il transforme en un nombre binaire. La deuxième technique, l'algorithme de Huffman, est un cas spécial d'algorithme arithmétique avec la seule différence qu'il transforme chaque symbole en un nombre binaire. Cependant, afin d'améliorer le taux de compression de ces deux algorithmes pour les images, de bons modèles prédictifs sont indispensables.

À la différence d'autres formes de données, la valeur d'un Pixel dans une image est fortement corrélée avec les intensités des Pixel environnants. Ceci a mené à la réalisation que le premier dérivé d'une image contient moins d'information que l'image de source [47]. Par conséquent, des techniques ont été développées comme la Modulation de Delta (MD) [48] et la Modulation du Codage d'Impulsion de Delta (MCID). Dans ces techniques, ce sont les différences entre les pixels consécutifs qui sont encodées plutôt que les valeurs des pixels elles-mêmes. Mais

avec la croissance continue de la taille des images, davantage compression était nécessaire.

Une technique plus efficace est employée sur les images pour fournir un contexte bidimensionnel pour chaque pixel. Cette approche permet à l'information contextuelle d'être appliquée dans le processus de prédiction ou de prévision des pixels. Pour améliorer les capacités prédictives du codeur, et ainsi le taux de compression, plus d'un Pixel précédent peut être employé pour la prédiction. Cependant, pendant que plus de Pixels sont employés, la performance tend à se dégrader. Les meilleurs algorithmes de compression sans pertes disponibles actuellement sont basés sur cette technique. Parmi ces algorithmes, on peut relever le codage adaptif contextuel sans perte des images (CALIC) [49]. Dans ce système 12 pixels sont employés pour prévoir la valeur du prochain et quatre évaluations de gradients locaux sont calculées à travers l'axe horizontal et vertical ainsi que les deux diagonaux. Même si ce système de compression sans perte est parmi les algorithmes les plus efficaces, il est rarement utilisé à cause sa complexité et sa grande consommation de temps. Le JPEG-LS [50] est un autre algorithme toujours dans la même famille de compression sans perte qui ressemble à CALIC mais plus efficace. Dans ce système seulement 4 pixels précédents sont employés dans la prédiction de la prochaine.

Même si tous ces algorithmes représentés ci-dessus fournissent des performances relativement bonnes pour la compression des images, ils n'ont pas réussi à réaliser des taux de compression très élevés. À la différence du texte, les images peuvent être légèrement tordues et ces changements sont imperceptibles dans le résultat final. Ceci signifie que le théorème de Shannon [41] sur les taux de compression n'est pas une obligation pour la compression des images. En effet, les algorithmes des compressions d'images qui n'obéissent pas au théorème de Shannon forment la deuxième famille d'algorithmes avec pertes d'informations. Les algorithmes de cette famille fournissent des taux de compression beaucoup plus élevés qui surpassent la performance de la majorité des algorithmes sans perte.

Algorithmes avec pertes d'informations :

La plupart des algorithmes de compression d'image avec pertes qui ont un taux élevé, sont basés sur l'utilisation de la transformation bidimensionnelle de l'image. Le but de cette transformation est de convertir les données de l'image de leur domaine spatial en un autre domaine (comme le domaine fréquentiel dans la transformée de Fourier), dans lequel les informations significatives sont réduites. Toujours dans le domaine de la transformée, on conserve les composantes qui contiennent ces informations significatives et on rejette les autres. Les résultats de ces transformations peuvent être convertis de nouveau au domaine original par l'application d'une transformation inverse. Par conséquent seules les transformées réversibles peuvent être utilisées.

La sélection des composantes peut être faite d'une façon simple par une sélection séquentielle de chaque composante jusqu'à ce qu'un niveau désiré de reproduction soit atteint. Cependant, en choisissant les composantes de cette façon on retient beaucoup de composantes qui ont peu ou pas d'effet sur l'image reproduites ou finale. Afin améliorer la performance de la sélection et du système de compression en général, le standard JPEG [51] utilise des tables de quantification pour quantifier ces composantes. Une fois les composantes sont quantifiées, celles qui dépassent un seuil prédéterminé sont conservées et les autres sont mises à zéro. Le nombre des composantes conservées est inversement proportionnel au taux de compression. Ceci étant dit, les transformées qui compressent toute l'information de l'image dans le plus petit nombre de composantes possibles est considéré comme une transformée optimale [25, 52].

La transformée qui fournit de tels résultats est la transformée de Karhunen Loève (TKL) [25]. Cependant, cette transformée est extrêmement complexe et comme sa fonction de base est déduite de l'image d'entrée, elle est inutile pour la compression des images.

La Transformée Discrète en Cosinus (TDC) est la meilleure transformée utilisée dans les algorithmes de compression d'images avec pertes et elle a démontré son efficacité et sa supériorité sur les autres transformées. Cela grâce à sa performance

asymptotique à celle de la transformée optimale de Karhunen-Loève [6, 25, 52]. Le standard JPEG est basé sur la TDC [51].

La Transformée Discrète en Cosinus appartient à une classe des transformées dont la fonction de base est orthogonale. Dans cette classe, on trouve d'autres transformées qui sont encore utilisées dans la compression d'image comme la Transformée Discrète de Fourier, la Transformée Discrète de Hartley et la Transformée Discrète de Walsh Hadamard et autres.

Une des nouvelles transformées récemment développées et appartenant à la famille des transformées orthogonales est la Transformée Discrète Fractionnelle de Fourier. Son application dans les différents domaines du traitement des signaux est en croissance rapide. En étudiant cette transformée, on a démontré qu'elle possède un grand potentiel dans la compression des images puisqu'elle a pu réaliser des taux de compression acceptable en l'appliquant sur des signaux unidimensionnels, exponentiels et autres [53].

Notre motivation dans ce qui reste de notre recherche est de tester la performance de la transformée fractionnelle de Fourier dans la compression des signaux bidimensionnels. Il restera à présenter les objectifs qu'on désire atteindre à la fin de cette analyse :

- 1- Trouver laquelle parmi les transformées orthogonales sous optimales étudiées dans les chapitres précédents est la plus optimale.
- 2- Évaluer la performance de la Transformée Discrète Fractionnelle de Fourier dans le domaine de compression de l'image.

8.2.Description de l'Algorithme :

La définition de la transformation que nous employons pour une image de $N \times N$ pixels est donnée dans les équations suivantes :

$$S[u, v] = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s[x, y] f[x, y, u, v] \quad (8.1)$$

$$s'[x, y] = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} S[u, v] g[x, y, u, v] \quad (8.2)$$

L'équation (8.1) représente l'image transformée $S[u, v]$, où u et v sont les coordonnées de l'image dans le domaine de la transformée. $s[x, y]$ est l'image, où x et y sont ses coordonnées dans le domaine normal. $f[x, y, u, v]$ est la fonction de base de la transformée utilisée. L'équation (8.2) donne la transformation inverse. La complexité de l'algorithme pour le calcul de la transformée d'une image de $N \times N$ est $O(N^4)$. Le calcul de la fonction de base f qui est formée de quatre dimensions est excessive et demande un grand espace dans la mémoire pour l'enregistrer. Les variables de toutes les fonctions de base qui vont être utilisées sont séparables. Ce qui signifie que la transformation bidimensionnelle peut être factorisée à une série de transformation unidimensionnelle en appliquant l'algorithme ligne colonne. Les équations (8.1) et (8.2) prendront les formes suivantes après la séparation de variables :

$$S[u, v] = \sum_{q=0}^{N-1} \sum_{p=0}^{N-1} f[u, p] s[p, q] f[q, v] \quad (8.3)$$

$$s[x, y] = \sum_{q=0}^{N-1} \sum_{p=0}^{N-1} g[x, p] S[p, q] g[q, y] \quad (8.4)$$

Dans les équations (8.3) et (8.4) on peut facilement démontrer que le nombre d'opérations exécutées pour la computation de la sortie est égale à $O(2N^3)$.

La procédure décrite ci-dessus peut être représentée dans l'algèbre linéaire comme une multiplication des matrices :

$$S = FSF^{-1} \quad (8.5)$$

$$s = GSG^{-1} \quad (8.6)$$

Où F et G sont les matrices équivalentes à f et g respectivement. En combinant les équations (8.5) et (8.6) on trouve :

$$s = GFSF^{-1}G^{-1} \quad (8.7)$$

L'équation (8.7) est valide seulement quand G est l'inverse de F et G^{-1} est l'inverse de F^{-1} . Comme toutes les fonctions de bases sont orthogonales alors il est facile de démontrer que $G = F^{-1}$. L'orthogonalité et la propriété de séparation des variables de

la fonction de base nous ont aidées à réduire la complexité de notre algorithme de $O(N^4)$ à $O(2N^3)$.

Encore, l'algorithme peut être amélioré en appliquant les algorithmes rapides introduites pour chaque transformée [9, 22, 48, 49] qui réduisent la computation d'un vecteur de N points de $O(N^2)$ à $O(N \log_2 N)$. Ceci rend la complexité de notre algorithme de traitement d'images à $O(2N^2 \log_2 N)$.

8.3.Méthode Expérimentale :

Dans cette section on discutera l'approche suivie pour comparer la performance de chaque transformée que l'on a étudiée. On commencera par une explication de l'étude réalisée et ensuite on passera à la présentation des paramètres adoptés pour mesurer la performance et la technique de sélection des composantes employées dans la reconstruction de l'image.

Pour effectuer une comparaison sur la propriété de compacité d'énergie de ces transformées, il était nécessaire d'analyser leur performance sur un certain nombre d'image de différentes natures. Pour cette raison on a choisi deux images qu'on va désigner par image de test #1 et image de test #2.

La première est la photo d'une personne et la deuxième est une panoramique du centre ville de Montréal. Ces images sont affichées dans la figure (8.1) et (8.2) respectivement.



Figure 8. 1: Image de test#1

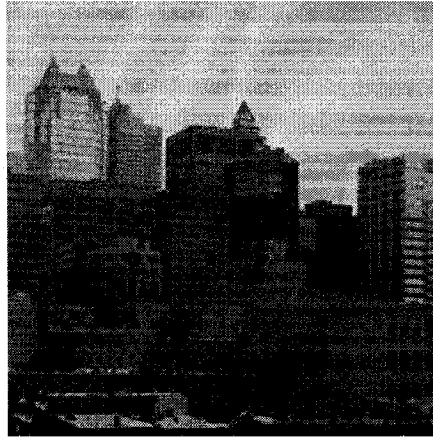


Figure 8. 2: Image de test#2

Chacune est une image de 256x256 pixels où l'intensité de chaque pixel est encodée par 8 bits dans l'échelle de gray.

Pour analyser et évaluer la performance de chaque transformée, on a calculé l'erreur dans l'image reconstruite pour différents nombres de composantes employées dans la reconstruction. Ces erreurs sont évaluées par le calcul de l'erreur quadratique moyenne (Mean Square Error) et le rapport signal maximal bruit (Peak Signal to Noise Ratio) qui sont définies par les équations (8.8) et (8.9) respectivement :

$$eqm = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I(i, j) - R(i, j)|^2 \quad (8.8)$$

$$PSNR = 20 \log_{10} \left(\frac{Max_I}{\sqrt{eqm}} \right) \quad (8.9)$$

Où $I(i, j)$ et $R(i, j)$ représentent les composantes de l'image originale et reconstruite respectivement. Max_I est le dynamique de l'image, dans notre cas est égale à 255 parce que les pixel sont codés par 8 bits. La performance est considérée bonne si la valeur d' eqm est petite et le $PSNR$ est grand.

Pour la sélection des composantes significatives, une procédure similaire à celle utilisée dans le standard JPEG [51] est adoptée. La procédure conserve les composantes qui sont plus grandes d'un seuil prédéterminé et rejette les autres. Par

cette procédure les composantes moins significatives dans l'image sont rejetées sans qu'elles n'introduisent un grand bruit dans l'image reconstruite.

La méthode présentée ci-dessus nous aide à comparer les transformées pour les propriétés de compacité d'énergie. Cependant, cette propriété n'est pas constante pour la Transformée Discrète Fractionnelle de Fourier. Elle varie lorsque l'ordre α de la transformée fractionnelle varie. Pour cela il est nécessaire d'effectuer notre analyse pour plusieurs valeurs de α .

8.4.Résultat et Analyse :

Dans cette section on discutera les résultats obtenus après la transformation des images mentionnées dans la section (8.3). On commencera par discuter les résultats individuels de chaque image et ensuite on finira par un résumé de résultats et une discussion générale.

Notre algorithme n'a pas inclus un système de quantification de composantes, ceci signifie que si toutes les composantes sont employées dans la reconstruction de l'image, il n'y aura pas d'erreur entre l'image originale et l'image reconstruite. Les graphes dans les figures (8.3) et (8.4) présentent les résultats de toutes les transformées pour l'image test#1 et l'image test#2 respectivement. Chaque graphe affiche la valeur de l'erreur (rapport signal maximal bruit) lorsque plus de composantes sont employées. Cela nous aide à évaluer la propriété de la compacité d'énergie de chaque transformée.

Pour que notre analyse soit crédible on a présenté dans le tableau (8.1) la valeur de l'erreur pour chaque transformée lorsque 20% des composantes sont employées dans la reconstruction des image.

Image de test#1		Image de test#2	
	PSNR (dB)		PSNR (dB)
TDC	96	TDC	89.9
TDH	94	TDH	89.7
TDF	90	TDF	86.2

TDFrF (0.9)	89	TDFrF (0.9)	85
TDFrF (0.8)	86.5	TDWH	84
TDWH	85	TDFrF (0.8)	82.5
TDFrF (0.6)	82	TDFrF (0.6)	76
TDFrF (0.4)	75	TDFrF (0.4)	70

Tableau 8. 1: Valeurs de l'erreur lorsque 20% des composantes sont employées.

Image de test#1 :

En examinant les résultats de cette image dans la figure (8.3) on peut aisément déduire que la Transformée Discrète en Cosinus (TDC) fournit la meilleure qualité de reconstruction, par conséquent la plus haute compacité d'énergie. La TDC est suivie par la Transformée Discrète de Hartley (TDH) et ensuite par la Transformée Discrète de Fouiner (TDF). La Transformée Discrète Fractionnelle de Fourier (TDFrF) et la Transformée Discrète de Walsh Hadamard (TDWH) ont une plus mauvaise performance. On remarque que la performance de la TDWH se situe entre ceux de la transformée fractionnelle d'ordre $(\alpha = 0.8\pi/2)$ et d'ordre $(\alpha = 0.6\pi/2)$.

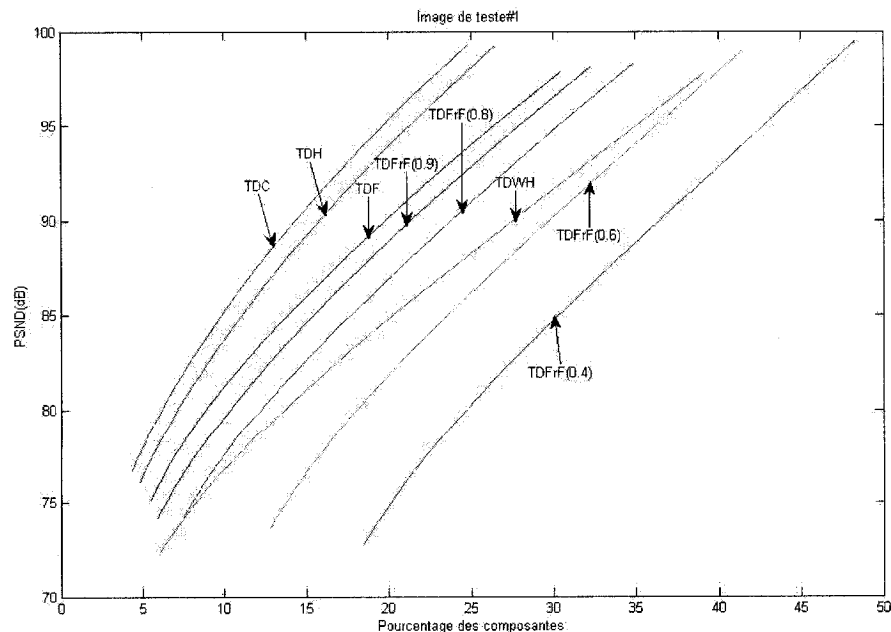


Figure 8. 3: l'erreur (PSNR) sur pourcentage des composants de transformées pour l'image de teste#1.

Image de test#2 :

Les résultats, figure (8.4), de l'image de test#2 dont la variation entre les pixels est plus grande que dans la première image révèle que la performance de toutes les transformées est un peu dégradée. La performance de la Transformée Discrète en Cosinus (TDC) est maintenant adjacente à celle de la transformée Discrète de Hartley (TDH) ceci signifie que la TDC est plus affectée par la dégradation que la transformée de Hartley.

Bien que la performance de la Transformée Discrète de Fourier soit un peu dégradée, elle a relativement maintenu son niveau de compacité d'énergie. Pour la Transformée Discrète de Walsh Hadamard on remarque qu'elle a fournit une meilleure performance dans cette image que dans l'image précédente. Sa performance est remontée pour être inscrite entre celle de la transformée fractionnelle d'ordre ($\alpha = 0.8\pi/2$) et d'ordre ($\alpha = 0.9\pi/2$).

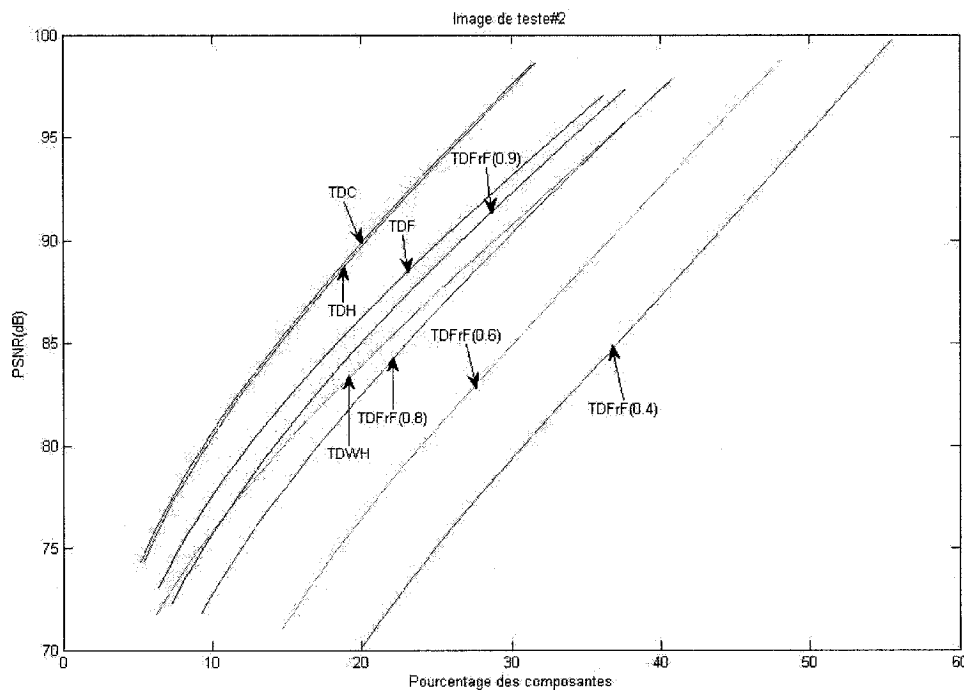


Figure 8. 4: l'erreur (PSNR) sur pourcentage des composants de transformées pour l'image de teste#2.

Résumé des résultats :

Après avoir examiné et analysé les résultats des transformées pour les deux images de test, on peut aisément conclure que la Transformée Fractionnelle et la Transformée de Walsh Hadamard sont moins performantes que les autres transformées. En comparant ces deux derniers à part, on remarque que la Transformée de Walsh est moins affectée par la dégradation produite sur la performance dans la deuxième que la Transformée Fractionnelle.

Quant à la Transformée Discrète en cosinus qui était la meilleure dans la première image, sa performance se dégrade pour se joindre à celle de la Transformée de Hartley dans la deuxième image. Finalement, la Transformée Discrète de Fourier a relativement maintenu sa position dans le palmarès de performance pour les deux images. Il faut noter ici que la courbe de la performance de la Transformée Discrète de Fourier est tracée en utilisant seulement la valeur absolue des composants. Par cela une partie de l'information que cette transformée fournit (la phase), était éliminée dans notre étude. Par conséquent, la performance de la transformée discrète de Fourier est un peu dégradée par rapport à sa performance calculée avec la phase.

8.5.Discussion:

Toutefois pas problématique pour cette analyse, il y avait un certain nombre de limitations qui ne peut ne pas être mentionné. Ces limites dévoilées ci-dessous élargissent les horizons de la recherche dans le domaine du traitement de signaux.

Sélection des composantes :

Cette limite s'encadre dans la méthode adoptée dans notre étude pour la sélection des composantes qui seront employées dans la reconstruction de l'image. Dans nos études, on a considéré les composantes dont la valeur dépasse un seuil prédéterminé comme les composantes significatives de l'image transformée. Par cette supposition, on n'a pas pris en considération la fiabilité du seuil choisi. Dans les techniques réelles les composantes sont quantifiées d'abord selon leurs importances dans l'image et après un seuil leur est appliqué. De cette façon là, le seuil peut être choisi adéquatement par rapport aux composantes. Notre algorithme

n'a pas intégré cette technique complexe qui nécessite un temps qui dépasse la portée de ce mémoire.

Analyse de la performance

Dans l'analyse de la performance adoptée, la seule limitation relativement signifiante dans cette étude est dans la façon avec laquelle l'erreur a été mesurée. L'erreur dans l'image reconstruite a été évaluée par des techniques mathématiques relativement primitives (*eqm et PSND*).

À la différence d'autres signaux, l'erreur dans les images est perçue mais n'est pas mesurée par l'œil humain, ce qui rend nos paramètres de mesure pas très efficace pour tester la qualité de l'image compressée. Dans le but de résoudre ce problème, une échelle de qualité des figures (Picture Quality Scale (PQS)) peut être utilisée [56]. L'échelle de qualité des figures donne une méthode de mesure plus objective pour quantifier la qualité de l'image reconstruite qui prend en considération plusieurs spécifications du système visuel humain.

Une autre technique introduite récemment [57] suggère que l'erreur dans les images doit être évaluée par le calcul du niveau de similarité structurale entre l'image originale et l'image reconstruite. Ceci provient du fait que le système visuel humain est plus sensible aux changements dans la structure de l'image.

Ces nouvelles techniques de calcul d'erreur se basant sur la quantification de la qualité visuelle de l'image reconstruite exige une période d'analyse relativement grande et leur implémentation pourrait dépassé la portée de cette étude.

Finalement, dans l'analyse et l'évaluation de la Transformée Discrète Fractionnelle de Fourier on a choisi arbitrairement les valeurs d'ordre α .

Pour une étude plus approfondie sur la propriété de compacité de l'énergie de cette transformée on devrait chercher l'ordre optimal qui fournit une haute décorrélation des informations de l'image. [53] a choisi le pourcentage de la racine quadratique moyenne de la différence (Percent Root mean square Difference PRD) comme paramètre pour trouver ce ordre optimal de la transformée fractionnelle.

9. Conclusion

Dans ce travail de recherche, une étude théorique sur les transformées orthogonales reliées à la Transformée Discrète de Fourier est faite. Ceci inclus en particulier, les transformées discrètes en cosinus, de Harley, de Walsh Hadamard et la transformée fractionnelle de Fourier. Dans notre étude, les définitions unidimensionnelles et bidimensionnelles ont été présentées, leurs propriétés discutées et une démonstration de leurs orthogonalités faite. Dans le but d'analyser leurs spectres obtenus après leur application à une classe des signaux, ces transformées ont été programmées à l'aide du logiciel Matlab.

Le rôle de la transformée de Fourier dans la reconnaissance de la parole a été étudié et discuté. Les limitations de la transformée traditionnelle de Fourier dans ce domaine sont démontrées. Des solutions qui résolvent ces contraintes, à l'aide des transformées reliées à Fourier ont été présentées.

Une comparaison d'optimalité est réalisée par une étude détaillée sur la propriété de compacité d'énergie de ces fonctions de bases sur les images. Les résultats de cette étude ont déjà été décrits dans cette recherche mais on les résume ci-dessous.

La Transformée Discrète Fractionnelle de Fourier et la transformée de Walsh Hadamard ont été moins performantes que les autres. En les comparant à part des autres, on a remarqué que la transformée de Walsh est moins affectée par la dégradation produite lorsqu'une image avec un grand niveau de variation entre les pixels adjacents est traitée.

Quant à la transformée discrète en cosinus qui était la meilleure dans la première image, sa performance s'est largement dégradée pour joindre celle de la Transformée Discrète de Hartley dans la deuxième image. Finalement, la Transformée Discrète de Fourier a relativement maintenu sa position au centre de ce palmarès de performance pour les deux images traitées.

10. Références

- [1] A. Srivastava, "Fourier Transforms As an Aid to Decision-Making" Technical Analysis of STOCKS & COMMODITIES, February 2005.
- [2] M. Yan, ZS. Lin, CT. Zhang, "A New Fourier Transform Approach for Protein Coding Measure Based on the Format of the Z Curve," *Bioinformatics*, 14(8), pp 685-690, 1998.
- [3]. S. Tiwari, S. Ramachandran, A. Bhattacharya, S Bhattacharya and R. Ramaswamy, "Prediction of Probable Genes by Fourier Analysis of Genomic Sequences," *Computer Applications in Biosciences* 13 (3), pp 263-270, 1997.
- [4] H.E. Chrestenson, "A Class of Generalized Walsh functions," *PACIFIC J. MATH.*, vol. 5, No. 1, pp 17-31, 1955.
- [5] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, vol. 23, pp 90, 1974.
- [6] K.R. Rao, P. Yip, "Discrete Cosine Transform; Algorithms, Advantages, Applications," Academic Press, Boston, 1990.
- [7] R. V. L. Hartley, "A More Symmetrical Fourier Analysis Applied to Transmission Problems," *Proc. IRE*, vol. 30, No. 3, pp 144–150, 1942.
- [8] R. N. Bracewell, "Discrete Hartley Transform," *J. Opt. Soc. Am.* vol.73, No. 12, pp 1832–1835, 1983.
- [9] R. N. Bracewell, "The Fast Hartley Transform," *Proc. IEEE* vol. 72, No. 8, pp 1010–1018, August 1984.

- [10] R. N. Bracewell, "The Hartley Transform," Oxford Univ. Press, New York, 1986.
- [11] V. Namias, "The Fractional Order Fourier Transform and its Application to Quantum Mechanics," J. Insr. Math. Appl., vol. 25, pp. 241-265, 1980.
- [12] A. C. McBride and F. H. Kerr, "On Namias' Fractional Fourier Transforms," /MA J. Appl. Math., vol. 39, pp. 159-175, 1987.
- [13] L. B. Almeida, "An Introduction to the Angular Fourier Transform," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 3, pp 257-260 (Minneapolis, MN), April 1993.
- [14] H. M. Ozatkas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, Filtering, and Multiplexing in Fractional Fourier Domains and their Relationship to Chirp and Wavelet Transforms," J. Opt Soc. Amer. A, vol. 11, No 2, pp. 547-559, February 1994.
- [15] M. J. Corinthios, "A time-series analyzer," Proc. Symp. Comput. Processing in commun., vol 19, MRI Symposia Ser. New York: Polytechnic press, pp. 47-60, 1969.
- [16] M. J. Corinthios, "The Design of a Class of Fast Fourier Transform Computers." IEEE Transactions on Computers, vol. C-20, No. 6, pp 617-623 June 1971.
- [17] M. J. Corinthios, "A Fast Fourier Transform for High-Speed Signal Processing," IEEE Transactions on Computers, Vol. C-20, No. 8, pp. 843-846 August 1971.
- [18] R. M. Haralick, "A Storage Efficient Way to Implement the Discrete Cosine Transform," IEEE Trans. Comput., vol. C-25, pp 764-765, June 1984.

- [19] Z. Wang, " Reconsideration of a fast computational algorithm for the Discrete Cosine Transform," IEEE Trans. Commun., vol. COM-31, pp. 121-123, Jan. 1983.
- [20] M. J. Corinthios, " Optimal Parallel and Pipelined Processing Through a New Class of Matrices with Application to Generalized Spectral Analysis," IEEE Transactions on Computers, vol. 43, No. 4, pp 443-459 April 1994.
- [21] K.R. Rao and P. Yip," On Shift Property of DCTs and DSTs," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol.35, pp-404, 1987.
- [22] L.N Wu " Comments on Shift Property of DCTs and DSTs," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol.38, pp-186, 1990.
- [23] F. R. Ohwsorg, "Binary Fourier representation," presented at the Spectrum Analysis Techniques Symp., Honeywell Res. Cen., Hopkins, Minn., Sept. 20-21,1966.
- [24] N. Ahmed, K.R. Rao, A.L. Abdussatar, " Bifore or Hadamard Transform" IEEE Trans. on audio and Electroacoustics, vol. AU-19, No. 3, pp 225-234, September 1971.
- [25] NU. Ahmed, KR Rao," Orthogonal Transforms for Digital Signal Processing" Springer-Verlag New York, NJ, 1975.
- [26] C. Yuen "Walsh Functions And Gray Code" IEEE Transactions on Electromagnetic Compatibility, vol. EMC-13, No. 3, pp 68-72, 1971.
- [27]Y.A. Geadah, M. J. Corinthios ' Natural, Dyadic, and Sequency order algorithms and processors for the Walsh Hadamard Transform ' IEEE Trans. on computers, vol. 26. No 5, pp 435-442, May 1977.

- [28] S. Pei and J. Ding "Closed-Form Discrete Fractional and Affine Fourier Transforms" IEEE Trans. on Signal Processing, Vol. 48, no 5, pp 1338-1353, May 2000.
- [29] H. M. Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdagi, "Digital Computation of the Fractional Fourier Transform," IEEE Trans. Signal Processing, vol. 44, pp. 2141–2150, Sept. 1996.
- [30] B. Santhanam, J. H. McClellan "The Discrete Rotational Fourier Transform," IEEE Trans. Signal Processing, vol. 42, pp. 994–998, Apr. 1996.
- [31] S. C. Pei, C. C. Tseng, and M. H. Yeh, "Discrete fractional Hartley and Fourier transform," IEEE Trans. Circuits Syst. II, vol. 45, pp. 665–675, June 1998.
- [32] S.C. Pei, M.H. Yeh, and C.C. Tseng, "Discrete Fractional Fourier Transform Based on Orthogonal Projection," IEEE Trans. Signal Processing, vol. 47, pp. 1335–1348, May 1999.
- [33] S. C. Pei and C. C. Tseng, "New Discrete Fractional Fourier Transform Based On Constrained Eigen-Decomposition of DFT Matrix by Lagrange Multiplier Method," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. vol. 5, pp. 3965–3968, 2005.
- [34] O. Arikan, M. A. Kutay, H. M. Ozaktas, and O. K. Akdemir, "The Discrete Fractional Fourier Transformation," In Proc. IEEE Int., Symp. Time-Frequency Time-Scale Anal., pp. 205–207, 1996.
- [35] M. S. Richman and T. W. Parks, "Understanding Discrete Rotations," in

Proc., IEEE Int., Conf., Acoust., Speech, Signal Process., vol. 3, pp.2057–2060, 1997.

[36] M.R. Portnoff, "Short-Time Fourier Analysis of Sampled Speech," IEEE Trans., Acoustics, Speech and Signal Processing, vol. ASSP-29, pp 364-373, 1981.

[37] R. Aaikara, Y. Gao and G Saon, "Fractional Fourier Transform Features for Speech Recognition," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing , vol. 5, pp 17-21, May 2004.

[38] R. Rozman, A. Strancar and D. Kodek, "Analysis of Window Function Influence on Robustness of Speech Recognition Systems," Proceedings of The ninth International Electro technical and Computer Science Conference ERK 2000, Portoroz, Slovenia, IEEE Region 8, vol. B, pp.177-180, Sept. 2000.

[39] R. Rozman and D. Kodek, "Improving Speech Recognition Robustness Using Non-Standard Windows," The IEEE region 8, Computer as a Tool, EUROCON'03, vol.2, pp 171-174, September 2003.

[40] C. Ris, V. Fontaine and H. Leich," Speech Analysis Based on Malval Wavelet Transform," Int. Conf. on Acoustics, Speech and Signal Processing, vol. 1, pp 389-392, May 1995.

[41] C. E. Shannon," A Mathematical Theory of Communication," Bell System Technical Journal, vol.27, pp 379–423 and pp 623–656, July and October 1948.

[42] J. Ziv and A. Lempel," A Universal Algorithm for Sequential Data Compression," IEEE Transactions on Information Theory, vol. IT-23, No. 3, pp 337–343, May 1977.

- [43] J. Ziv and A. Lempel, "Compression of Individual Sequences Via Variable Rate Coding," IEEE Transactions on Information Theory, vol. IT-24, No. 5, pp 530–536, September 1978.
- [44] L.-J. Kau and Y. P. Lin "Lossless Image Coding Using a Switching Predictor With Run-Length Encodings" IEEE International Conference on Multimedia and Expo (ICME), vol.2, pp 1155-1158, June 2004.
- [45] J.C David. MacKay, "Information Theory, Inference, and Learning Algorithms," Cambridge University Press, Chapter 6, pp-115, September 2003.
- [46] D.A. Huffman, "A Method for The Construction of Minimum-Redundancy Codes", Proceedings of the of the Institute of Radio Engineers, vol 40, No.9, pp 1098-1102, September 1952,
- [47] D. Chen and A. C. Bovik," Visual Pattern Image Coding," IEEE Transaction on Communications, vol. 38, No. 12, pp 2137–2146, 1990.
- [48] A.N. Netravali and J.O Limb,"Picture Coding: A Review," IEEE Proceedings on image processing, vol.68, No.3, pp 366–406, 1980.
- [49] X. Wu and N. Memon," Context-Based, Adaptive, Lossless Image Coding (CALIC)," IEEE Transactions on Communications, vol. 45, No. 4, pp 437–444, 1997.
- [50] J. M. Weinberger, G. Seroussi, and G. Sapiro, "FROM LOCO-I TO THE JPEG-LS STANDARD," Proceedings of the IEEE int. conference on Image processing, Kobe, Japan, Vol. 4, pp 68–72, October 1999.
- [51] G. K. Wallace, "The JPEG Still Picture Compression Standard," Commun. ACM, vol. 34, No. 4, pp 30–44, 1991.

- [52] A. K. Jain, "Fundamentals of Digital Image Processing". Prentice-Hall, 1998.
- [53] C. Vijaya, J.S. Bhat, "Signal Compression Using Discrete Fractional Fourier Transform and Set Partitioning in Hierarchical Tree," *Signal Processing*, vol. 86, pp1976–1983, 2006
- [54] J.W. Cooley and J.W. Tukey, "An Algorithm for The Machine Calculation of Complex Fourier Series" *Math of Comput.*, vol. 19, pp 297-301. 1965.
- [55] W.H. Chen, C. H. Smith, and S.C. Fradrick, "A Fast Computation Algorithm for The Discrete Cosine Transform" *IEEE trans. Commun.*, vol. COM-25, pp 1004-1009, Sept. 1977.
- [56] R. Algazi, M. Miyahara, and K. Kotani. "Objective Picture Quality Scale (PQS) for Image Coding," In *Proceedings of the Society for Information Display International Symposium*, volume 23, pages 859–862, 1992.
- [57] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.
- [58] M. J. Corinthios, K. C. Smith, J.L. Yen," A Parallel Radix-4 fast Fourier Transform Computer," *IEEE Transactions on Computers*, Vol. C-24, No. 1, pp 80-92, January 1975.
- [59] M. J. Corinthios," Generalised transform factorization for massive parallelism," *IEEE proc, Vis. Image Signal Process.*, vol. 151, No. 3, pp153-163, June 2004.

[60] H. M. Ozaktas, Z. Zalevsky, M. A. Kutay," The Fractional Fourier Transform with Applications in Optics and Signal Processing," John Wiley & Sons Ltd., Chapter 3, pp 93-99

[61] X. Yifang, Z. Jinjie, Y. Kaisheng, C. Zhigang, M. Zhengxin," Robust Recognition of Noisy Speech using Speech Enhancement," IEEE proc. int. conf. signal processing, vol. 2, pp 734-737, Aug 2000.

11. Annexe#1: Programme du Matlab

Le programme produit pour le calcul et la comparaison des transformées pour la compression des images est présenté ci-dessus.

Le programme de Matlab :

```
% lecture de l'image

I=imread('IMG_2161.png');
II=I(1:256,1:256);
I1=double(I);
I2=double(II);
N=256;
K=257;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformée de Fourier
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e_dft=zeros(1,25);
MSE_dft=zeros(1,25);
PSNR_dft=zeros(1,25);
mse_dft=zeros(1,25);
% la matrice directe

for k=1:K;
    kk=k-N/2;
    for n=1:K;
        nn=n-N/2;
        dft_f(k,n)=exp(-i*2*pi*(nn-1)*(kk-1)/N);
    end
end

% la matrice inverse

for n=1:K;
    nn=n-N/2;
    for k=1:K;
        kk=k-N/2;
        dft_i(n,k)=exp(i*2*pi*(nn-1)*(kk-1)/N);
    end
end
F=1/K*dft_i;
% transformation
```

```

DFT=dft_f*I1*F;
U_dft=abs(DFT);

% compression

for t=1:25
for y=1:K;
    for x=1:K;
        if U_dft(x,y)<t;
            Z_dft(x,y)=0;
            e_dft(t)=e_dft(t)+1
        else
            Z_dft(x,y)=U_dft(x,y);
        end
    end
end

% calcule de l'erreur quadratique moyenne

for y=1:K;
    for x=1:K;
        MSE_dft(t)=((U_dft(x,y)-Z_dft(x,y)))^2+MSE_dft(t);
    end
end
mse_dft(t)=1/(K^2)*MSE_dft(t);

% Calcule du rapport signal sur bruit

PSNR_dft(t)=20*log(255/sqrt(mse_dft(t)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformée en Cosinus
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e_dct=zeros(1,25);
MSE_dct=zeros(1,25);
PSNR_dct=zeros(1,25);
mse_dct=zeros(1,25);

% la matrice directe
for k=1:N;
    for n=1:N;
        if k==1

```

```

        w(k)=1/sqrt(N);
    else
        w(k)=sqrt(2/N);
    end
    dct_f(k,n)=w(k)*cos((pi*((2*n)-1)*(k-1))/(2*N));
end
end

% la matrice inverse

for n=1:N;
    for k=1:N;
        if k==1
            w(k)=1/sqrt(N);
        else
            w(k)=sqrt(2/N);
        end
        dct_i(n,k)=w(k)*cos((pi*((2*n)-1)*(k-1))/(2*N));
    end
end

% transformation

DCT=dct_f*I2*dct_i;
U_dct=abs(DCT);

% compression

for t=1:25;
    for y=1:N;
        for x=1:N;
            if U_dct(x,y)<t;
                Z_dct(x,y)=0;
                e_dct(t)=e_dct(t)+1
            else
                Z_dct(x,y)=DCT(x,y);
            end
        end
    end
end

% calcule de l'erreur quadratique moyenne
for y=1:N;
    for x=1:N;
        MSE_dct(t)=(abs(DCT(x,y)-Z_dct(x,y)))^2+MSE_dct(t);
    end
end

```

```

end
mse_dct(t)=1/(N^2)*MSE_dct(t);

% Calcule du rapport signal sur bruit

PSNR_dct(t)=20*log(255/sqrt(mse_dct(t)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Transformée de Hartley
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e_dht=zeros(1,25);
MSE_dht=zeros(1,25);
PSNR_dht=zeros(1,25);
mse_dht=zeros(1,25);

% la matrice directe

for n=1:K;
    nn=n-N/2;
    for k=1:K;
        kk=k-N/2;
        dht_f(n,k)=cos((2*pi*(nn-1)*(kk-1))/K)+ sin((2*pi*(nn-1)*(kk-1))/K);
    end
end

% la matrice inverse

dht_i=1/K*dht_f;

% transformation

DHT=dht_f*I1*dht_i;
U_dht=abs(DHT);

% compression

for t=1:25;
    for y=1:K;
        for x=1:K;
            if U_dht(x,y)<t;
                Z_dht(x,y)=0;
                e_dht(t)=e_dht(t)+1
            else

```



```

        Z_dht(x,y)=DHT(x,y);
    end
end
end

% calcule de l'erreur quadratique moyenne

for y=1:K;
    for x=1:K;
        MSE_dht(t)=(abs(DHT(x,y)-Z_dht(x,y)))^2+MSE_dht(t);
    end
end
mse_dht(t)=1/(K^2)*MSE_dht(t);

% Calcule du rapport signal sur bruit

PSNR_dht(t)=20*log(255/sqrt(mse_dht(t)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformée de walsh Hadamard
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e_wht=zeros(1,25);
MSE_wht=zeros(1,25);
PSNR_wht=zeros(1,25);
mse_wht=zeros(1,25);

% generation de la matrice directe

p=log2(N);
h=[1 1;1 -1];
H=1;
for i=1:p
    H=kron(h,H);
End
% la matrice inverse
wht_i=1/N*H;

% transformation
DWT=H*I2*wht_i;
U_wht=abs(DWT);

% compression

```

```

for t=1:25;
for y=1:N;
    for x=1:N;
        if U_wht(x,y)<t;
            Z_wht(x,y)=0;
            e_wht(t)=e_wht(t)+1
        else
            Z_wht(x,y)=DWT(x,y);
        end
    end
end
end

% calcule de l'erreur quadratique moyenne

for y=1:N;
    for x=1:N;
        MSE_wht(t)=(abs(DWT(x,y)-Z_wht(x,y)))^2+MSE_wht(t);
    end
end
mse_wht(t)=1/(N^2)*MSE_wht(t);

% Calcule du rapport signal sur bruit

PSNR_wht(t)=20*log(255/sqrt(mse_wht(t)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformée Fractionnelle de Fourier
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MSE_frac=zeros(4,25);
PSNR_frac=zeros(4,25);
mse_frac=zeros(4,25);
N_frac=128;
K_frac=(2*N_frac)+1;
Afrac=sqrt(1/(2*N_frac+1));
e_frac=zeros(4,25);
a=[0.4 0.6 0.8 0.9];

% la matrice directe

for b=1:4;
ang=a(b)*pi/2;
p=sqrt(2*pi)*cos(ang)/K_frac;
s=1;

```

```

for u=1:K_frac;
    uu=u-N_frac;
    for v=1:K_frac;
        vv=v-N_frac;
        frac_f(u,v)=Afrac*exp((i*(uu-1)^2*p)/2)*exp(-i*s*2*pi*(vv-1)*(uu-1)/(2*N_frac+1))
                    *exp((i*(vv-1)^2*p)/2);
    end
end

```

% la matrice inverse

```

p1=-p;
s1=-s;
for r=1:K_frac;
    rr=r-N_frac;
    for t=1:K_frac;
        tt=t-N_frac;
        frac_i(t,r)=Afrac*exp((i*(rr-1)^2*p1)/2)*exp(-i*s1*2*pi*(tt-1)*(rr-1)/(2*N_frac+1))
                    *exp((i*(tt-1)^2*p1)/2);
    end
end

```

% transformation

```

ffrac=frac_f*I1*frac_i;
U_frac=abs(ffrac);

```

% compression

```

for t=1:25;
    for y=1:K_frac;
        for x=1:K_frac;
            if U_frac(x,y)<t;
                Z_frac(x,y)=0;
                e_frac(b,t)=e_frac(b,t)+1
            else
                Z_frac(x,y)=U_frac(x,y);
            end
        end
    end
end

```

% calcul de l'erreur quadratique moyenne

```

for y=1:K;
    for x=1:K;

```

```
        MSE_frac(b,t)=((U_frac(x,y)-Z_frac(x,y)))^2+MSE_frac(b,t);
    end
end
mse_farc(b,t)=1/(K^2)*MSE_frac(b,t);

% Calcule du rapport signal sur bruit

PSNR_frac(b,t)=20*log(255/sqrt(mse_frac(b,t)));
end
end
```